

PHOTOPLETHYSMOGRAPHIC SENSOR-BASED NON-INTRUSIVE AND
SECURE SMART SENSING AND APPLICATIONS

by

A B M Mohaimenur Rahman

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Computing & Info Systems

Charlotte

2022

Approved by:

Dr. Yu Wang

Dr. Pu Wang

Dr. Weichao Wang

Dr. Dong Dai

©2022
A B M Mohaimenur Rahman
ALL RIGHTS RESERVED

ABSTRACT

A B M MOHAIMENUR RAHMAN. Photoplethysmographic Sensor-based Non-intrusive and Secure Smart Sensing and Applications. (Under the direction of DR. YU WANG)

In the past few decades, there have been revolutionary developments in the field of smart sensing. The new era of next-generation intelligent systems is leveraging the usage of smart sensing technology to perform intelligent sensing tasks and collect useful information for different applications. Smart sensors, besides the task of collecting information from an object and converting it into an electric signal, can facilitate different diagnoses, functions, identifications, and conclusion-oriented tasks after processing the signal using advanced signal processing and artificial intelligent algorithms. The advancement of smart sensing techniques and applications is advancing rapidly and has been implemented in modern mobile devices or wearable devices. Different smart sensing systems use different types of sensors, such as temperature, pressure, infrared, proximity, light, acoustic, motion, magnetic, and vibration. This dissertation discusses secure smart sensing and applications based on the non-intrusive Photoplethysmography (PPG) sensor, which is commonly available in current wearable devices.

There are different aspects of secure smart sensing systems, such as purely security-based solutions to protect smart sensing and secure applications built based on smart sensors. Our focus of this dissertation is the latter one. In this dissertation, we first study how to authenticate a user's offline/online signature with data from the PPG sensor. Conventionally, offline signatures are verified manually in the banks, and for the online signatures, dedicated electronic devices like tablets are used for verification. We propose a novel method for both offline and online signature authentication, which leverages the widely deployed PPG sensors in the wrist-worn wearable devices. The unique blood flow changes in the supplicant's hand movement are being exploited in

this system to validate the signature. We design a low-cost hardware implementation to verify our proposed method. Our experiments with real-life data sets verify the feasibility and efficiency of the proposed solution.

In addition, we also study a smart application of PPG sensing for weight lifting assessment. Physical activity (PA) plays an important role in a person's health. Weight lifting is one of the essential stationary exercises which helps a person maintain a fit lifestyle. It is also important for a person to be aware of the intensity of performed exercise during a workout. In our proposed work, the PPG-based system is able to classify a user's lifted weighted object into its corresponding weight label. It leverages the change in the blood volume in the wrist region that occurred due to the strain caused by the different weights being lifted in order to classify the labels. We believe the importance of PPG sensing in secure smart sensing and applications during this technology era is immense.

ACKNOWLEDGEMENTS

I want to express my gratitude to God Almighty, The Most Merciful, before I start acknowledging anyone. Without His blessings, it would not have been possible.

Firstly, I would like to acknowledge my advisor Dr. Yu Wang without whose continuous mentoring, support, and help, I could not have completed my dissertation. He is the best advisor one could possibly have. I am really fortunate to work under his supervision and learn countless things that will keep me running for the rest of my life.

I am grateful to all the members of my dissertation committee for their constructive feedback, advice, and help in improving my dissertation overall. I also like to acknowledge the tuition support provided by the Graduate School and the financial support provided by the Department of Computer Science throughout my Ph.D.

I want to thank my parents, family, and friends for their constant support. Finally, I want to especially thank my wife Moon for her continuous support, care, affection, sacrifices, and also for helping me out during the data collection process of different experiments. Her consistent belief in me, encouraging me and handling my mental stress kept me going.

TABLE OF CONTENTS

LIST OF TABLES	x
LIST OF FIGURES	xi
LIST OF ABBREVIATIONS	xiv
CHAPTER 1: INTRODUCTION	1
CHAPTER 2: BACKGROUNDS	6
2.1. Smart Sensing	6
2.2. Secure Sensing Applications	7
2.3. Photoplethysmography Smart Sensing	9
2.4. PPG Sensor-based Secure Applications	11
2.5. Sensor-based Fitness Applications	13
CHAPTER 3: PPG-BASED OFFLINE/ONLINE SIGNATURE AUTHENTICATION (POSA)	17
3.1. Introduction	17
3.2. Problem Definition and Security Model	18
3.2.1. Signature Authentication Problem	18
3.2.2. Security Model	20
3.3. System Design	21
3.3.1. System Architecture	21
3.3.2. Workflow	23
3.4. Challenges	24
3.4.1. Coarse-grained Wrist PPG Signals	24
3.4.2. Different PPG Readings of Same User	25

3.4.3.	Effect of PPG Sensor Placement	26
3.4.4.	Overlapped Signature Signals	27
3.5.	Data Segmentation	28
3.5.1.	Pre-processing	29
3.5.2.	Signature Segmentation	31
3.5.3.	Signature Separation	35
3.6.	Feature Characterization	37
3.6.1.	Feature Extraction	38
3.6.2.	Relevant Features	38
3.6.3.	Feature Matrix	43
3.6.4.	Classifiers	44
CHAPTER 4: POSA IMPLEMENTATION AND EVALUATION		45
4.1.	Experimental Setup	45
4.1.1.	POSA Band	45
4.1.2.	Server	46
4.2.	Data Collection	47
4.3.	Evaluation	48
4.3.1.	Metrics	49
4.3.2.	Performance	49
CHAPTER 5: PPG-BASED WEIGHT LIFTING ASSESSMENT (PaWLA)		55
5.1.	Introduction	55
5.2.	Problem Definition	56

5.3. System Design	58
5.3.1. System Architecture	58
5.3.2. Workflow	60
5.4. Challenges	61
5.4.1. Coarse-grained Wrist PPG Signals	61
5.4.2. Effect of PPG Sensor Placement	62
5.4.3. Similarity in Readings for Nearby Weights	62
5.5. Data Processing	63
5.5.1. Noise Filtering	63
5.5.2. Signal Normalization	64
5.5.3. Signal Trimming	64
5.6. Feature Characterization	64
5.6.1. Feature Extraction	65
5.6.2. Feature Selection	65
5.6.3. Feature Matrix	66
5.6.4. Classification	67
CHAPTER 6: PAWLA IMPLEMENTATION AND EVALUATION	68
6.1. Experimental Setup	68
6.1.1. PaWLA Band	68
6.1.2. Weight Accessories	69
6.1.3. Server	69
6.2. Data Collection	69

	ix
6.3. Evaluation	70
6.3.1. Metrics	71
6.3.2. Performance	71
CHAPTER 7: CONCLUSION AND FUTURE WORKS	76
7.1. Conclusion	76
7.2. Future Works	77
REFERENCES	79

LIST OF TABLES

TABLE 3.1: List of Extracted Features by <i>tsfresh</i> .	39
TABLE 3.2: Continued List of Extracted Features by <i>tsfresh</i> .	40
TABLE 3.3: Continued List of Extracted Features by <i>tsfresh</i> .	41
TABLE 3.4: Continued List of Extracted Features by <i>tsfresh</i> .	42
TABLE 3.5: List of Relevant Features selected by <i>tsfresh</i> .	43
TABLE 6.1: Distribution of Weights per Category.	70
TABLE 6.2: Performance of User Independent Model (80% training).	75

LIST OF FIGURES

FIGURE 1.1: Overview of the dissertation.	4
FIGURE 2.1: General architecture of smart sensing systems.	7
FIGURE 2.2: PPG readings for cardiac cycles.	10
FIGURE 3.1: A generalized signature authentication problem.	19
FIGURE 3.2: POSA security model.	20
FIGURE 3.3: System architecture of POSA.	22
FIGURE 3.4: Workflow of POSA.	24
FIGURE 3.5: Example of PPG data from fingertip & wrist [1].	25
FIGURE 3.6: Example of a user with different PPG pulse signal.	26
FIGURE 3.7: Location for placement of the PPG sensor.	27
FIGURE 3.8: A schematic representation of POSA's signal overlapping.	28
FIGURE 3.9: A schematic representation of data segmentation module flow.	29
FIGURE 3.10: Comparison after applying noise filtering method.	30
FIGURE 3.11: Comparison after applying data normalization method.	31
FIGURE 3.12: Segmentation of signature signal using Skewness-DTW method.	32
FIGURE 3.13: Segmentation of signature signal using dynamic programming method.	33
FIGURE 3.14: Segmentation of signature signal using binary segmentation method.	34
FIGURE 3.15: Feature characterization module flow.	37
FIGURE 4.1: Overall hardware setup for data collection via POSA band.	46

FIGURE 4.2: Data collection using the POSA band prototype.	47
FIGURE 4.3: Performance of POSA for different segmentation algorithms averaged over all classifiers with 60% training data.	50
FIGURE 4.4: Performance of POSA for different training size varying from 20%-80% averaged over all classifiers.	51
FIGURE 4.5: Performance of POSA for different classifiers (Skewness-DTW segmentation with 80% training data).	52
FIGURE 4.6: Comparison of the signature portion of the three different locations of PPG-sensor placement (a) median antebrachial vein (b) radial artery (c) posterior wrist-side.	53
FIGURE 4.7: Performance of POSA on the three different surfaces (a) wood (b) metal (c) plastic.	54
FIGURE 4.8: Performance of POSA under attacks from the same user.	54
FIGURE 5.1: PPG sensor readings for lifting weights of 4 lb and 5 lb.	57
FIGURE 5.2: A generalized strain sensing problem.	58
FIGURE 5.3: System architecture of PaWLA.	59
FIGURE 5.4: Workflow of PaWLA.	60
FIGURE 5.5: Flow in feature characterization module.	65
FIGURE 6.1: PaWLA prototype and data collection: (a) hardware components of PaWLA Band; (b) weight accessories; (c) data collection with a PaWLA band.	69
FIGURE 6.2: Overall performance of classification for (a) even and (b) odd weights.	72
FIGURE 6.3: Performance of PaWLA with (a) different classifiers (weight category of 2 lb difference, 60% data as training data) (b) different training sizes varying within 20 – 80% of data (average among all weight categories).	72

FIGURE 6.4: Performance of PaWLA for (a) different sensor placements (60% as training data) (b) different weight differences (80% as training data). 73

FIGURE 6.5: Performance of user specific model for each user (80% training). 74

LIST OF ABBREVIATIONS

AS-LMS Adaptive Step-size Least Mean Square

CA Continuous Authentication

COTS Commercially Off The Shelf

CSI Channel State Information

DTW Dynamic Time Warping

ECG Electrocardiogram

GB Gradient Boosting

IMU Inertial Measurement Unit

IoT Internet of Things

k-NN k Nearest Neighbor

MA Motion Artifacts

MLP Multi Layer Perceptron

MOSA Multi-modal Offline/Online Signature Authentication

NN Neural Network

OvR One vs Rest

PA Physical Activity

PASS PPG-based Application for Strain Sensing

PaWLA PPG-based Weight Lifting Assessment

PIN Personal Identification Number

POSA PPG-based Offline/Online Signature Authentication

PPG Photoplethysmography

RBF Radial Basis Function

RF Random Forest

RPE Rating of Perceived Exertion

RTFS Real-Time Feedback System

S-BSS Semi-Blind Source Separation

S-G Savitzky-Golay

SVM Support Vector Machine

WSN Wireless Sensor Network

CHAPTER 1: INTRODUCTION

The proliferation of smart devices equipped with built-in sensors is enabling a new paradigm called smart sensing [2]. Smart sensing is an emerging field for research in this exponentially evolving tech era. Almost most of the households are turning into smart homes equipped with smart devices. Devices now-a-days are equipped with multiple sensors to sense data from their surroundings in order to perform a specific action or task according to the need of the user. The use of sensors and communication technology and the processing of the sensed data using advanced signal processing and artificial intelligent algorithms in combination form Smart sensing. Applications based on smart sensing is used nowadays in almost every sector to reduce human intervention and increase automation. Mainly it is used for various monitoring and control systems.

In this world of emerging technology, as days go by the number of smart devices thus the number of sensors is increasing. In these smart devices or in the smart sensing systems, the use of different types of sensors, such as temperature, pressure, infrared, proximity, light, acoustic, motion, magnetic, and vibration are found. These sensors can be either passive or active in action. When it measures the effect of the energy in the environment for a purpose it is passive. On the other hand, when the sensors themselves emit some energy and measure the response of the environment based on that energy it is the active scenario. Each sensor modality has its own advantages which are exploited in diversified applications. The range of these applications is vast. Fields such as, transportation, traffic automation, health monitoring, agriculture, telecommunication, military, industrial machinery, industrial logistics, education, home automation, and offices. The popularity of smart sensor systems is

on the rise because of its higher advantages over its price. Besides being low-cost, these systems are generally highly reliable with high performance and implementable with easy maintenance.

This dissertation is based on one of the commonly available sensors called Photoplethysmography (PPG) Sensor found in wearable devices. It discusses secure smart sensing and applications made with the help of this non-intrusive PPG sensor. The focus is primarily on the secure applications or smart applications built on PPG sensor rather than purely security-based solutions to protect smart sensing.

A Photoplethysmogram is an optical signal reading to measure the changes in the blood volume under the skin. There are mainly two phases of a cardiac cycle which are diastole and systole, relaxing and contracting of the heart muscles respectively to run the blood flow throughout the body. These blood flow reaches the terminal part of the body such as, fingers, ear lobes, palms, and wrists. PPG sensors are composed of an LED light and an optical receiver i.e. a photo-diode. The LED light is transmitted towards the skin which illuminates the skin and then the photo-diode captures the reflected light from the skin as the optical reading. The optical readings reflect the cardiac cycles with peaks if there is no movement of the body part associated with the sensor. And if there are movements of the body parts then the output readings reflect the movements in its signals. PPG sensors are used in mainly medical devices for a wide variety of applications such as, measurement of Heart rate and its variability, Blood pressure, Blood oxygen saturation, assessing respiratory conditions, detection of arterial diseases, and monitoring of various health conditions. These are also found to be in the smart wearable devices such as, smart watches and fitness trackers to monitor pulse rate and other health aspects.

And as the use of these devices and sensors is increasing, the risk of maintaining user's security is also increasing [3]. Issues related to privacy invasion, breach of personal information, email scamming, hacking, phishing, are on the rise [4, 5, 6, 7].

To solve these issues, various methods like PIN numbers, passwords, certificates, biometrics have been used [8]. In recent times, different types of authentication schemes [9, 10, 11, 12, 13, 14] are proposed in the literature. The main purpose is to verify a user before the user starts to use the device or product. On the other hand, attackers try different methods to by-pass these schemes in order to access the system or device to fetch or breach the user's privacy. It is a competition between the attackers and the researchers trying to come up with new techniques and strategies to attack and defend a system respectively. Existing solutions might have tried solving these security issues but it was observed that the solutions were not that secure or non-intrusive. Some of them require user extra involvement which seriously affects user experience and delay authentication time. Few physiological-based techniques are inconvenient for users to authenticate frequently and continuously. There are some systems which can verify users non-intrusively but are not that effective against different advanced attacks like statistical, replay, hacking, etc. But PPG sensors on the other hand can provide non-intrusive and secure authentication.

Hence, in our first work, we propose a system which tries to verify online/offline signatures of a user leveraging the PPG sensed data from the wrist-worn wearable. Current scenario in banks or financial institution is that the user's signature has to be manually verified or if it is online i.e. on a tablet or dedicated smart device it has to be verified using various software module which may suffer from various common attack model mentioned above. To make the solutions convenient, the use of existing wrist-worn wearable devices come into play. Almost all the smartwatch/fitness watches have this PPG sensor as a built-in sensor. Mainly all types of smartwatch/fitness watches make use of PPG sensors to measure user's heart rate.

The other work is another smart sensing application based on PPG sensor for strain sensing where a user wearing a smartwatch or fitness tracker is able to recognize the certain amount of weight being lifted based on the strain occurring when he/she is

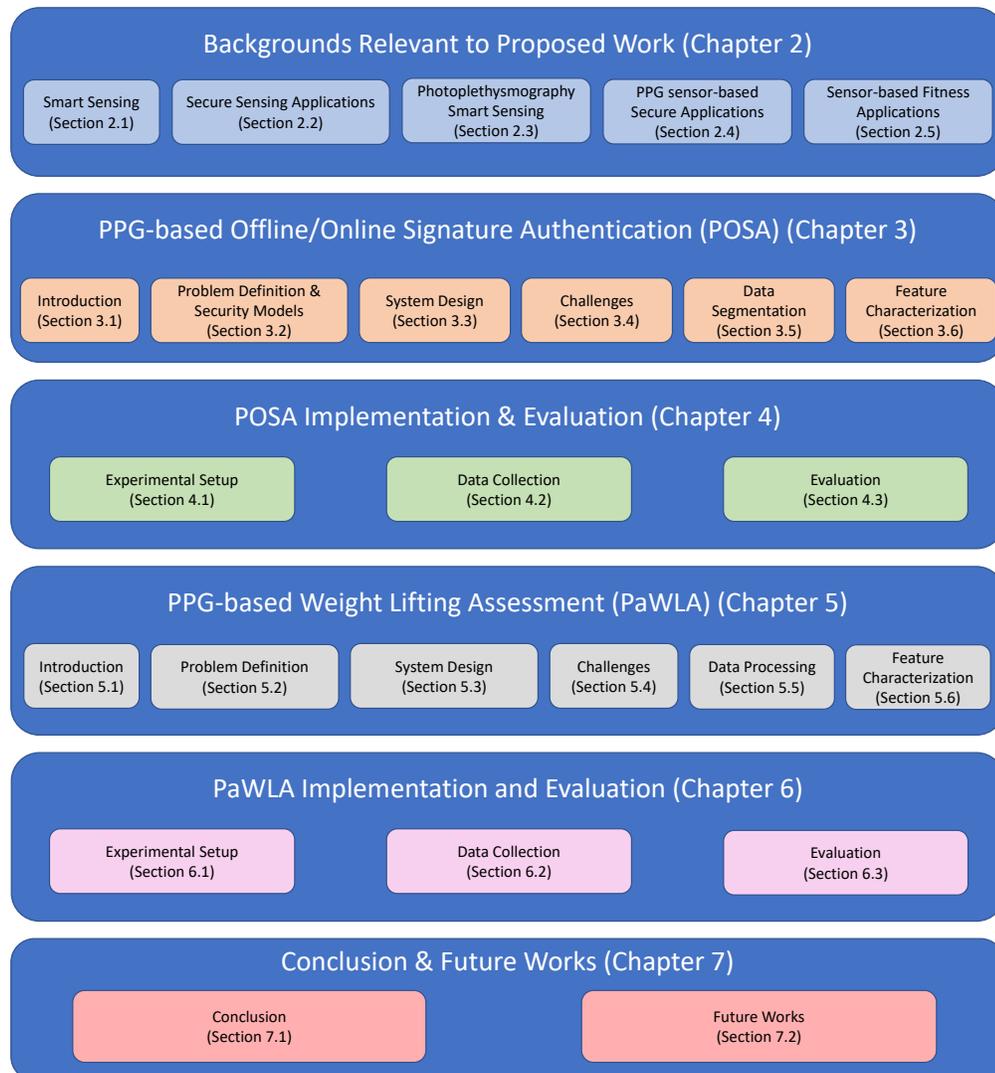


Figure 1.1: Overview of the dissertation.

lifting the weight.

The first work regarding the signature authentication can be used in banking sectors, agreement signing scenarios or in any smart device sign in. It could be used in situations where an authentication of signature, whether it is online or offline, is needed. For the strain sensing work, there are different potential applications. For example, automatic detection of weights when doing weight-based workouts, and recognizing the weight of objects outside of workout-scenario. Besides these applications,

there are further potential in scenarios where the awareness of the weight lifting is required.

In summary, in this dissertation, we mainly focus on a secure signature authentication system and a smart sensing system both based on PPG sensing data. The overview of this dissertation is shown in Fig. 1.1. The rest of this dissertation is organized as follows. We first introduce the backgrounds and related works to the dissertation in Chapter 2. The above-mentioned first work will be introduced in Chapter 3 along with the technical details. Its implementation and evaluation will be presented in Chapter 4. The strain sensing work mentioned earlier will be introduced in Chapter 5. The implementation and evaluation of the strain sensing work is discussed in Chapter 6. Finally, Chapter 7 concludes the dissertation and presents the potential future works.

CHAPTER 2: BACKGROUNDS

In this chapter, we provide brief backgrounds on related work in smart sensing, secure sensing applications, PPG-based smart sensing, PPG sensor-based secure applications, and sensor-based fitness applications.

2.1 Smart Sensing

The sensing paradigm where smart devices equipped with smart sensors are used is called smart sensing. In this fast changing technology era, there have also been revolutionary developments in the smart sensing area. Households are becoming smart homes consisting of a wide range of smart devices. The devices are using the smart sensing technology to perform different intelligent sensing tasks. Different diagnoses, functions, identifications, conclusion-oriented tasks are also performed after processing the raw signals received by the smart sensors using advanced signal processing and artificial intelligent algorithms. Smart-sensor based applications are making human life easier and comfortable in all sectors. Smart sensing applications are now an integral part of Internet of Things (IoT). A general architecture of a smart sensing system is shown in Fig. 2.1.

A variety of sensors is incorporated in different smart sensing systems. Sensors such as infrared, proximity, temperature, pressure, light, acoustic, accelerometer, motion, magnetic, vibration, and air quality are commonly found in the existing systems [15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26]. Based on how it operates, these sensors can be two types - active and passive. When the sensors transmit and measure the reflections from the environment or object then it is an active sensor. And when the sensor just directly measures the sensing element from the object or environment it is

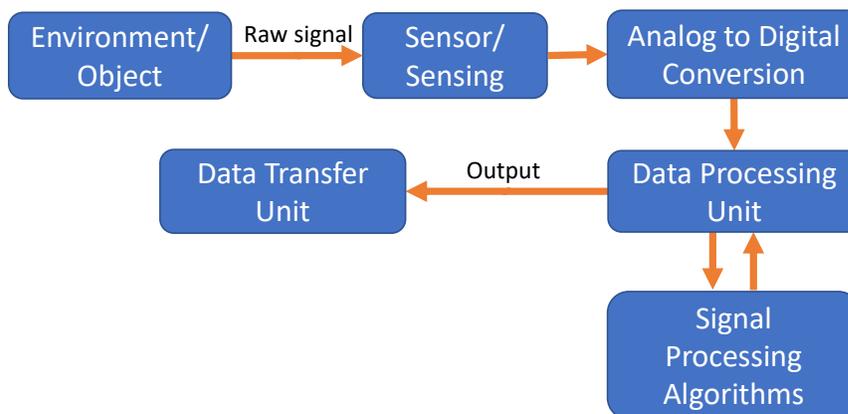


Figure 2.1: General architecture of smart sensing systems.

called a passive sensor. The popularity of smart sensing systems is increasing because of the following advantages -

- **low-cost:** The cost of sensor devices has been continuously reduced, which make them more popular in current mobile devices;
- **highly reliable:** Smart sensors can monitor the health of a system and control any failure or fault which makes them highly reliable in different scenarios.
- **high performance:** The computational overhead is really less for smart-sensors compared to the actions performed by them.

The applications are spread over almost every fields because of the popularity such as transportation, traffic automation, health monitoring, agriculture, military, industrial machinery, industrial logistics, education, home automation, and offices [27, 28, 29, 30, 31, 32, 33, 34, 35].

2.2 Secure Sensing Applications

The requirement for technological devices and applications is a reliable and convenient authentication process. Traditional user authentication mechanisms lead to

security vulnerabilities and have serious usability issues. This influences the user to just leave their devices unprotected while trying to simplify their authentication processes. In ideal case, the authentication process should be such that it is easy, fast and convenient to the user but susceptible to the attacker. To meet these requirements, various sensor-based authentication schemes came into the picture. In the current days, almost all the smart phone devices contain fingerprint sensors for authentication. Fingerprint as a bio-metric for authentication scheme is very popular [36]. Last few years have been a great advancement in the camera sensor-based face recognition schemes for authentication. Even though face recognition is fast and intuitive, there are still limitations from computer vision perspective [37]. Camera is also used in smartphones to capture the unique characteristics of a cardiac cycle to detect cardiac motion patterns. These unique characteristics are leveraged to authenticate a user [12]. People find that camera-based solutions are a threat to privacy. Lu *et al.* in [38], used acoustic sensing to read the user's lips or mouth movement in order to extract unique patterns to verify the users using a deep learning-based method in smartphone. In the literature, a lot of the systems are using the built-in sensors such as, accelerometer, gyroscope, and magnetometer in order to authenticate users. For example, in [39], Ehatisham-ul-Haq *et al.* developed a user authentication framework where the behavioral traits of smartphone users is exploited after the data of the embedded sensors (accelerometer, gyroscope, and magnetometer) was processed.

Another area of secure applications is the verification of signatures. There are two types of signatures out of which the first one is the offline and the other one is the online signature. When users give their signature on a piece of paper with a pen it is offline signature. And when the signatures are given on an electronic device or any smart device/hardware it is called the online signature. Different sensors have been used in the existing systems in order to verify offline/online signatures. For example, in [40], Wang *et al.* used a pen-type device where there were force sensors

to measure the force signal between the pen tip and the paper. After collecting the force signals, they match them with the saved signals using Dynamic Time Warping (DTW) method for verification. Bromley *et al.* came up with a time delay neural network called "Siamese" where there were two identical sub-networks joined together at their outputs and their setup contained pen-input tablet [41]. Most commonly used sensors in the literature for signature verification are accelerometer, and gyroscope. Shastry *et al.* designed a custom pen hardware consisting of accelerometer, gyroscope, temperature, and magnetometer sensors to collect sensor data capturing dynamic information of the signature such as, instantaneous acceleration, number of maxima and minima, rotation, rotation time series and then the extracted features are stored and a combination of dynamic time warping and hidden Markov models with Gaussian mixtures is used as the classifier [42].

Secure application such as intruder detection is also an area which is being focused in the recent era. Arjun *et al.* in their survey paper, explained the wireless sensor network (WSN) techniques related to intruder detection and border surveillance [43]. The applications based on the WSN techniques mainly focus on different sensors such as, seismic, acoustic, light, and surveillance camera. In [44], Al-qaness *et al.* introduced a device-free intruder detection and alarm system where they leveraged the channel state information (CSI) of COTS Wi-Fi to detect intruders. Alsalami *et al.* also introduces a comprehensive study to detect intruders using visible light communication [45] .

2.3 Photoplethysmography Smart Sensing

Photoplethysmogram is an optical measurement method which measures the changes in the blood volume in tissue level. And the changes in the blood volume generally occur due to cardiac cycles. A cardiac cycle helps run the blood flow through out the human body. It consists of diastole and systole which are relaxation and contraction of the heart muscles. As it flows throughout the body, it reaches the terminal parts

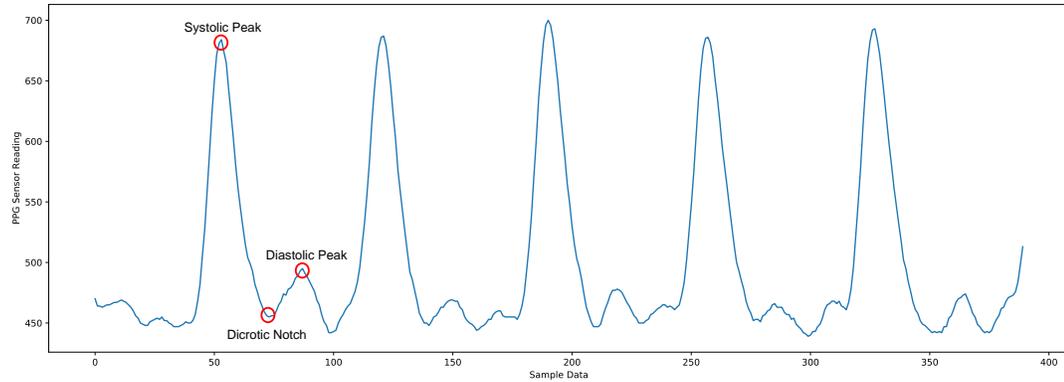


Figure 2.2: PPG readings for cardiac cycles.

of our body like, fingers, ear lobes, palms, and wrists. The optical method of Photo-plethysmogram consists of light transmission via an optical LED and reception of the reflected energy via a photo-diode. Mainly, it is getting the reading from just below the surface of the skin where the blood change can be detected. PPG is considered as a non-invasive technology. When plotted, the PPG sensor reflects the systolic and diastolic peaks in its readings as shown in Fig. 2.2. The second derivative of this PPG signal reading contains additional health-related information such as cardiovascular illness can be evaluated by analysing the waveform [46]. In general for the LED, green color LED is selected, as it has the highest absorption capacity for both oxyhaemoglobin and deoxyhaemoglobin with respect to other light sources [47].

The use of PPG sensor is frequently found in physiological measurements. Applications such as, measurement of blood pressure [48], heart rate and its variability [49], blood oxygen saturation [50] or assessing and detection of respiratory and arterial diseases [51], and monitoring of different health conditions. Over the past few years, the health monitoring technologies have moved to wearable devices. And PPG sensor is preferred over ECG sensor-based systems because of the simpler hardware implementation, lower costs, and the requirement of only one sensor to measure the data. PPG sensors are generally placed at some easily accessible anatomical positions of the body such as, wrist, ear lobes, fingertip, and forehead.

The forehead region is a good place for PPG sensing as it has a thin skin and a lot of blood vessel passing all around it. This is the reason which reduces the effect of motion artifacts by increasing the quality of the PPG signal reading [47]. Mendelson *et al.* had experimented on forehead by placing an array of sensors mounted on a soldier's helmet to find out that the the signal is less noisy when the pressure between the sensors and tissues is minimal [52]. One of the most frequently placed positions is the earlobe for PPG devices. The ear lobes contain a large number of blood sources. Poh *et al.* used an earring type setup for placing the PPG sensor on the earlobes [53]. In [47], the PPG sensor was installed inside an earphone and earbud for collecting the reading. The PPG sensor can also be placed inside the ear canal where the signal is less noisy compared to other location of the ear [54]. The most popular location to place a PPG sensor is the wrist position. According to [47], wrist-type PPG devices are popular because (1) wristband-type devices are inexpensive; (2) they are highly portable; and (3) users find those convenient to wear.

In [55], Lee *et al.* placed the PPG sensor around the radial and ulnar arteries in the wrist. Thomas *et al.* in [56], proposed a wristband-type PPG device called "BioWatch" where there are two electrodes on the bottom of the device in contact with the arm the device is worn on and there is another electrode on the opposite side i.e. on the top of the device making contact with the other arm using any finger.

2.4 PPG Sensor-based Secure Applications

This section talks about the contribution of Photoplethysmography sensor based systems to conduct reliable authentication process. PPG-based systems are efficient, reliable, secure and non-intrusive. Some of the systems use oximeters to collect the data and some use wrist-worn wearable which has the built-in PPG sensor [57, 58, 59, 60, 8, 61, 62, 63, 64, 1]. As wrist-worn wearable devices are on the increase and most of them contains the PPG sensor, the research community is starting to understand the importance of PPG-based smart sensing. Now, there is a great interest in biometric

authentication on a smartwatch due to the fact that it is worn and is with the user almost all the time. There are also some works based on other sensors embedded in the smartwatches/fitness trackers e.g. motion sensors, accelerometer, gyroscope, etc. But the problem of these systems are that a good amount of space is needed for the user to perform certain gestures and thus take a lot of time for the whole authentication process. So, it is not feasible for scenarios where fast authentication is required or the user is not in a space to perform certain gestures.

In [65], Choudhary *et al.* proposed a novel Normalized Cross Correlation (NCC) based PPG biometric method which is noise-robust and applicable for body area networks and m-health applications. The method firstly pre-processes the PPG signal then the systolic peaks are detected. After that, the averaged pulsatile waveform is extracted and finally similarity matching is done based on NCC measure. *Karimian* used an adaptive quantization approach to extract PPG biometric-based key generation [66]. Firstly, a wavelet transformation is done and then from the output, the reliable PPG features are only considered for the key generation phase. *Yathav et al.* developed a custom made handheld device where they collected PPG signals besides ECG signals as a robust hardware for on-the-go biometric identification [67]. For their comparative analysis they considered heart rate variability as their basis. In [60], *Kavsaoglu et al.* considered running a feature ranking algorithm over the time domain features extracted from the first and second derivative of the PPG signal and passing it to a k-NN classifier model for the final identification.

As the popularity of wrist-worn wearable is increasing rapidly, research are being done on wearable comprising PPG sensors. *Spooren et al.* in [57], implemented a multi-factor authentication where at first they extracted an rPPG signal from the facial video for liveness check and then finally the PPG signal is extracted from the user's smart watch. *Zhao et al.* for the first time introduced a low-cost Continuous Authentication (CA) system by extracting PPG signal from wrist-worn wearable

which represents the cardiac biometrics of an user [1]. In their system, they developed such mechanisms which effectively identify and eliminate motion artifacts (MA) and generate fiducial features to capture the uniqueness of the users' cardiac patterns. An adaptive gradient boosting tree based classifier was used to continuously authenticate the users. In [64], Cao *et al.* besides presenting a two-factor authentication system based on PPG sensing, also presented a two stage Motion Artifacts (MA) removal algorithm to separate clean heartbeat signals. Their system does not have the restriction to the user to stay still during the process of authentication. They also have a backup authentication method where if the biometrics fail, there is a repeatable and non-invertible method to generate cancelable feature templates as alternative credentials.

Though the PPG sensor measures the volumetric change in the blood flow, most of the existing works consider extracting the pulse signal for further processing to authenticate the users. Each work has its own set of features extracted and tried implementing disparate classifiers to finally identify the legitimate users from the illegitimate ones. In the first part of this dissertation, we will use PPG sensors for signature authentication, which has not been done before.

2.5 Sensor-based Fitness Applications

So far we have talked about the backgrounds on smart sensing, secure sensing applications, PPG smart sensing, PPG sensor-based secure applications. For the final section of the background chapter, we will be talking about sensor-based fitness applications. Over the recent years, studies have been conducted related to monitoring and assessing stationary exercises using smart sensing techniques.

In order to recognize the type of exercises or physical activity, the number of repetitions, sets, and phases, plenty amount of research has been conducted so far. Qi *et al.* [68] proposed a two-layer recognition framework where stationary exercises, aerobic, and free weight activities and sets & repetitions of free weight exercises were

classified. Two Shimmer3 wireless wearable sensors were placed on the subject's wrist, and chest respectively. For sedentary activities, the chest is an ideal measurement position as it is closer to the center of body mass and also the heartbeat can be obtained. They also placed a sensor on the wrist to increase recognizer accuracy as arm movements play an important role in most of the physical activities. In [69], Pernek *et al.* implemented a hierarchical algorithm which is composed of two layers of Support Vector Machines (SVMs). The first layer is to recognize which type of exercises is being performed. And the second layer recognizes the intensity of the exercise. A single SVM is used in the first layer to sever the purpose. After the type of exercises is recognized, based on it, the second layer's SVM predicts the corresponding intensity which is mainly the intensity prediction for the previously predicted type of exercise. The system focuses on a set of upper-body exercises using different weight loads. Their system consists of COTS five wearable sensors and one smartphone connected over Bluetooth into a piconet local area network. These systems had used multiple wearable sensors as did [70] and [71]. Most recently, several sensing systems [72, 73, 74], leveraging a single wearable sensor in one location, have been proposed to determine the type of training exercises and identify sets, repetitions, and phases in a workout.

There are also infrastructure-based solutions besides these wearable systems where sensors such as cameras, RFID readers, WiFi access points are deployed in the environment passively instead of directly on the objective body. For example, an acoustic based personalized fitness monitoring system to classify fitness actions and also identify users along side was proposed by Xie *et al.* in [75, 76] where a smart speaker and a microphone array are used for active acoustic sensing. Another example is a work from Xiao *et al.* in [77] where back-scattered wireless signals are used that are obtained at passive tags to detect and/or recognize macro/micro human movements during the exercises.

Besides the work on physical activity and performance recognition, there have also been many works on the qualitative analysis of exercises using camera-based, wearable device-based, or wireless device-based methods. It is very common for systems to use Kinect in *camera-based methods*. Works such as [78, 79, 80, 81] leveraged Kinect for monitoring fitness activities and demonstrating the users on how to improve the performance. Ai *et al.* in [78], developed a real-time feedback system (RTFS) using a Kinect sensor to capture the depth data. They also video-recorded the sessions and digitized the data by the SIMI Motion system under the same reference system. The comparisons of the displacements were almost similar. The feedback provided by the system was like the maximal height of the bar, the trajectory of the bar, and so on. They analyzed the data to track the movement of barbell. Yasser *et al.* [79] leveraged an IR camera (Microsoft Kinect Xbox 360) to detect the misplaced joints of the athlete while doing a lift, and alert the athlete before an injury can occur. They used the Fast Dynamic Time Warping (FastDTW) method to detect if the lift was correct or wrong and to detect what type of mistake has been made in the lift. They worked on Shoulder Press, Deadlift, and Squat and focused on 6 joints for each of them. 10 athletes aged from 20-30 years participated in the experiments. For most of the *wearable device-based methods*, one or multiple wearable devices are carried by the users during the exercises, and the sensing data from those wearable devices are analyzed to understand the quality of the exercises performed by the users. Velloso *et al.* [82] focus on the qualitative assessment of the exercise and providing feedback to the user. They investigated three aspects - specifying correct execution, detecting execution mistakes, and giving feedback to the user. They have used 3 types of sensors - accelerometer, magnetometer, and gyroscope along the 3 axes and in 4 positions. So, a total of 12 sensors were used for the dataset. They developed two systems - one with ML techniques where they collected data from participants while they performed exercises correctly and with different types of mistakes and the other one

with a model-based approach. In Weight-Mate [83], a prototype wearable system was designed for giving weightlifters of different skill levels personalized, precise and non-distracting immediate feedback on how to correct their current body positioning during deadlift training. They developed a wearable suit Weight-Mate suit consisting of 14 IMU sensors. They generated a 3D model in UNITY based on the sensor readings. They gave audio, visual, and textual feedback to the weightlifters based on their postures while lifting the weights. Similarly, iCoach [84], FitCoach [85] and GymSoles [86] can assess the quality of the workout leveraging a smart fitness glove, smartwatch/smartphone mounted on upper arms, and an insole prototype, respectively. Different wireless technologies such as acoustic, RFID, Wi-Fi can also be used for assessing exercises qualitatively in case of *wireless device-based methods*. For example, SEARE [87] leverages the Channel State Information (CSI) received from Wi-Fi devices to evaluate the quality of the exercises.

One of the important features in the commonly available wearable devices such as wristbands, smartwatches, and fitness watches is monitoring the heart rate during a workout or even during daily routine activities. This is achieved by the contributions from PPG sensor. But in our second work of this dissertation, different from all these research, we investigate the quantitative side of a weight lifting activity. Our proposed system focuses on the automatic recognition of the weight lifted by a user using only a single wearable PPG sensor. To the best of our knowledge, we are the first to address this specific quantitative analysis problem of a weight lifting activity via smart sensing.

CHAPTER 3: PPG-BASED OFFLINE/ONLINE SIGNATURE AUTHENTICATION (POSA)

3.1 Introduction

Fraudulent activities persist all over the world among which financial fraud is the major one. Despite various prevention methods from the financial institutions, paper checks continue to lead the way of transaction and they are susceptible to fraudulent attacks. Besides paper checks, other important legal and financial documents still require handwritten signatures to verify a person. And attackers try to forge those signatures in order to by pass the system to conduct their fraudulent activities.

To help prevent these fraudulent activities, different types of authentication systems have been proposed in the literature depending on the type of signature. Mainly, there are two types of handwritten signatures - offline and online. Offline signatures are the ones user give on a piece of paper with a pen. And online signatures are the ones user give on an electronic device which might be a tablet or any other smart device/hardware. Almost all the offline signature verification methods use image processing in their systems to identify the correctness of the signature [88, 89, 90, 91, 92]. And for online signature verification, first the data is collected from the electronic device on which the user is giving his/her signature to be authenticated. Then the signature data is sent to be pre-processed and based on generated features a model is trained which finally decides whether the signature is valid or not [93, 94, 95, 96, 97].

In this work, we have proposed a novel method for both offline and online signature authentication which leverages the widely deployed PPG sensors in the wrist-worn wearable devices. There have been some works regarding user authentication based on PPG sensors. However, there are no existing works that deal with offline or online

signature verification. The unique blood flow changes in the supplicant’s finger and hand movement is being exploited in this system to validate the signature. The motivation behind the idea are the limitations of existing works which are:

- Relies on dedicated devices [98, 99, 100, 101, 102, 103].
- Finger-worn devices are limited to gestures of a specific finger [104, 105, 106].
- Uses motion data of wearable to authenticate signatures [107, 108, 109].

These methods enjoy the low cost, non-intrusiveness, and easy deployment. However, many of them still require users’ extra effort (for calibration or training) and suffer from low accuracy (due to environmental noises or different attacks). Therefore, there is still a need for new low-cost, non-intrusive, pervasive, robust signature authentication methods.

3.2 Problem Definition and Security Model

In this section, we first define the signature authentication problem and then introduce the security model we used.

3.2.1 Signature Authentication Problem

The signature authentication problem is primarily a user verification problem but based on signature. The handwriting analysis is a close field related to this problem where the users’ handwriting is analysed to recognize different written letters or gestures. In [110], Ardüser *et al.* proposed a framework which could recognize text data collected from a smartwatch when writing on a whiteboard. Xu *et al.* had a system where the users wrote on a sheet of paper and the platform would infer the letters being written [111]. Signature can also be considered a type of handwriting as the user is giving the signature using their hand. So, it can be said that the problem here is authenticating a user after analysing the handwritten signature either on a sheet of paper or on a smart device.

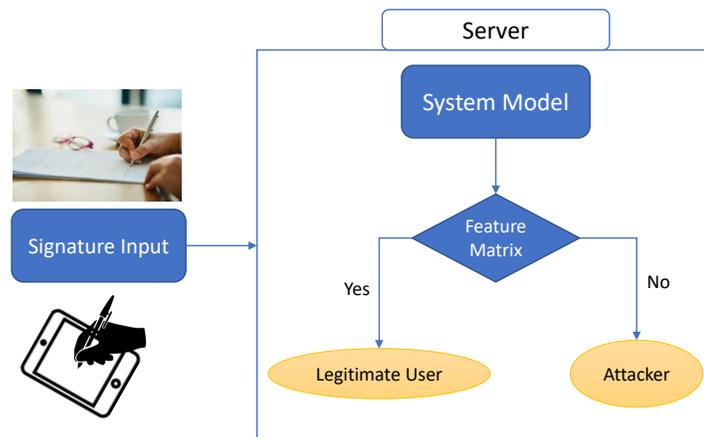


Figure 3.1: A generalized signature authentication problem.

The problem of signature authentication is like any other authentication problem in the field of security. Typically, a user gives his/her signature online/offline to claim to be a certain person. The system should be such that it would be able to verify the claim whether the user is a legitimate user or an illegitimate user. The data for the signature can be collected in a lot of ways such as, using additional hardware (e.g., cameras, signature pads, or custom-built sensors) or leveraging the commercially off-the-shelf (COTS) smart devices (e.g., smartwatches or wristbands).

In our work, the data is the sensor reading of Photoplethysmography sensor and is collected from the wrist-worn wearable device which contains the PPG sensor. The trajectory of the used pen affecting the blood flow of the users' wrist is taken into consideration. The effect is reflected upon the PPG reading samples over the time. Thus the signal can be represented as: $S(n)$ where $n = 1, 2, 3, \dots, N$ and each of the samples are separated at a time interval of $1/100$ Hz or 0.01 seconds.

A user first registers into the system with his/her own signature. The system is trained on that registered signature to create a model based on the features extracted from that signature. Then when a new signature input is given into the device, the system model based on the saved feature matrix decides whether the user is a

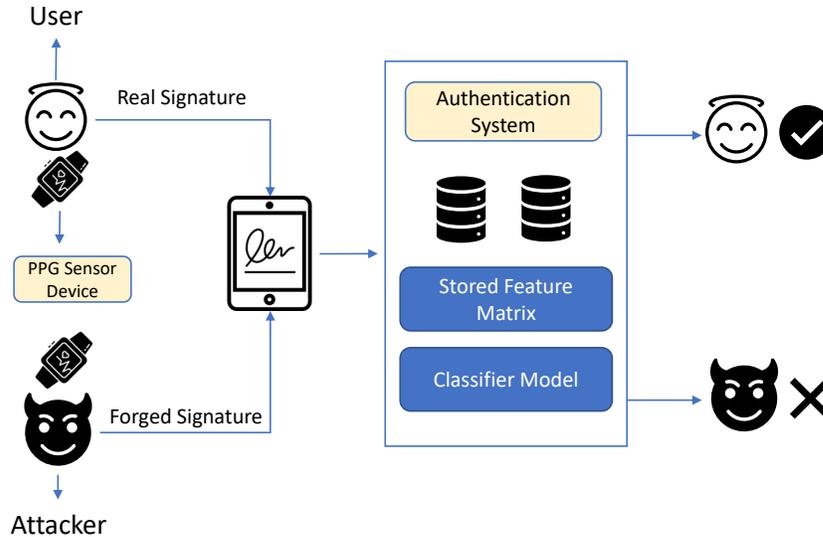


Figure 3.2: POSA security model.

legitimate user or an attacker. Many consider it as a machine learning classification problem. A general picture of the process can be seen in Fig. 3.1. In case of PPG-based system, the input to the system model would be the PPG sensor's data.

3.2.2 Security Model

As it is an authentication system, there will always be trusts and threats within the system model. Our system tries to prevent fraudulent activities in scenarios where an illegitimate user is trying to forge signature to claim to be a different person. The whole security model consists of four main entities, which are mentioned below with their trust assumptions:

- **Users:** This is the authorized or legitimate person who should be the only one to be approved by the system. A user registers his/her signature in the system wearing the smart watch with the PPG sensor built-in. The system trains on those registered signatures to create a model which later verifies the user.
- **Smart Devices:** A smart device has the PPG sensor as a built-in sensor to collect the PPG readings while the user/attacker gives the signature. It is

generally a wrist-worn wearable device which is also suitable for our system. Also, for collecting the online signatures, another device is considered where the signature is given. In this work, it is assumed that the smart devices are trustworthy.

- **Server & Links:** After completing the writing of the signature, the PPG sensor data from the smart device is sent to the server for data processing and verification. We assume that the server itself and the communication link between the devices and the server are all trustworthy.
- **Attackers:** An attacker tries to forge the signature of the legitimate user to bypass the system to do fraudulent activities. We assume that the attacker can come close to the user and shoulder peek to observe the signature and the way the user gives his/her signature.

The security model and how the authentication system operates to protect the user is shown in Fig. 3.2. As you can see, the attacker tries to forge the signature like the real user. Both the sensing data of the real user and attacker goes into the authentication system but only the real user is approved and the attacker is rejected by the system.

3.3 System Design

Now we introduce the overall system architecture and workflow of POSA.

3.3.1 System Architecture

The system architecture of POSA is shown in Fig. 3.3, which consists of four parts: *Data Collection*, *Data Segmentation*, *Feature Characterization*, and *Classification*. The wrist-worn wearable device is used to collect the signature data written either on a sheet of paper or on a smart device screen in the data collection module. After the collection of the data it is passed on to the data segmentation module where there are

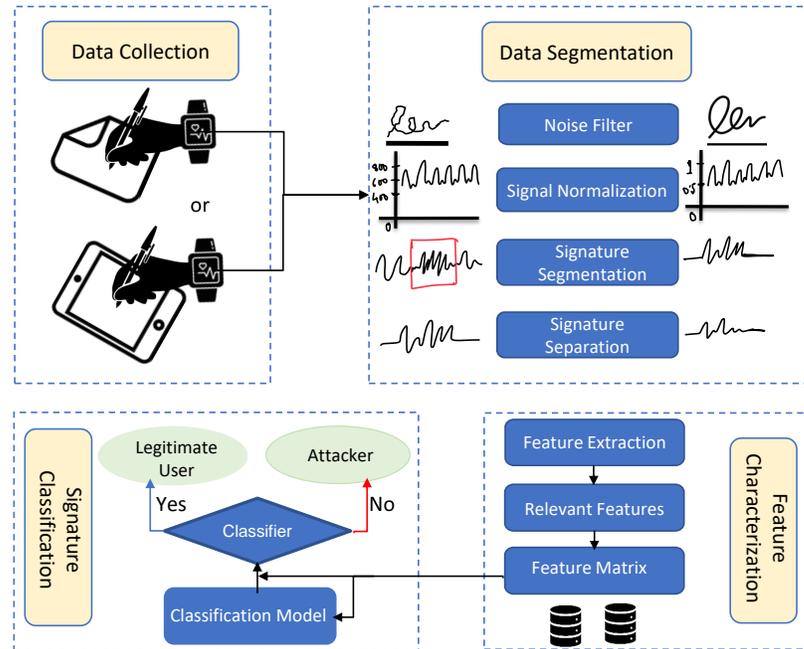


Figure 3.3: System architecture of POSA.

four steps of processing: noise filter, signal normalization, signature segmentation, and signature separation. Noise filter has the ability to filter out the combined noises of the signal. The signal normalization component normalizes any signal that comes into it. Two very important components are signature segmentation and separation where there are algorithms to segment the signature from the main data and then separate the overlapped signature and pulse signal data. After the data segmentation module, comes the feature characterization module which includes three main components: feature extraction, relevant features, and feature matrix. This module mainly deals with the extraction of relevant intrinsic characteristics of the input signal data that can discriminate each user's signature from another. Finally, the last module is the classification module which is the foundation model of the system. The classifier model is trained on the whole dataset of a user in order to classify a new incoming signal into any of the two categories: legitimate user or attacker.

3.3.2 Workflow

The workflow of POSA is illustrated in Fig. 3.4. There are primarily two main phases for the system which are: *Training Phase*, and *Authentication Phase*. The workflow of the system in light of the phases is discussed below.

Training Phase: In the training phase, every new user provides some sample signatures of his/hers using their wrist-worn wearable device. The provided PPG sensor signal data is then sent to the server-end from the user-end for each new user. After the data collection whose details are discussed in Section 4.2, the data goes to the Data Segmentation module. The noise from the raw data is first filtered & normalized and passed on to the next step for segmentation & separation. The raw data contains a mixture of pulse signal and signature motion artifact. In the segmentation stage, the signature portion is segmented from the raw PPG data. Even though we segment the signature portion, it contains a combination of both pulse signal and signature artifact which is separated in the signature separation step. After receiving the processed signature data, features are extracted and filtered according to relevance in the feature characterization module mentioned in the previous Section 3.3.1. Feature matrix is created based on the relevant features and stored in the database for each user. The classifier is trained based on the feature matrix to create a model for each user. The whole process can be related in analogy with the process of registering a new vehicle under a user. The user provides all the necessary information to the DMV to register his/her vehicle. Later if there is any scenario where the vehicle needs to be inspected for example, if police stops the vehicle to check then all the previous stored information is used to verify the user as the owner of the vehicle.

Authentication Phase: In the authentication phase, the data collection is done in the same way as in the training phase. The user trying to gain access, would provide the signature while wearing the wrist-worn wearable device. The data would then be sent to the server-end for data processing which includes the noise filtering of

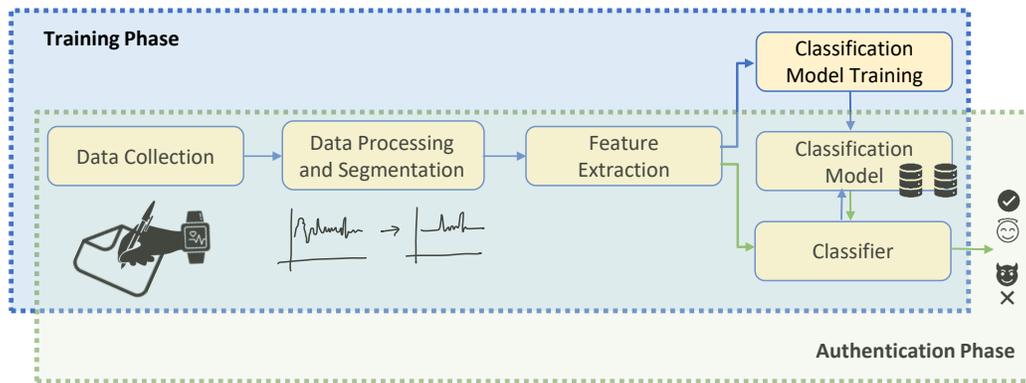


Figure 3.4: Workflow of POSA.

the raw data, normalizing the data, segmenting the signature portion from the whole collected signal, and finally separating the signature source from the combination of signature and pulse signals. After that, the feature matrix is generated based on the same feature extraction process mentioned in the training phase. Now, this feature matrix is passed on to the pre-trained classifier model for validation of the user. Then finally, the system lets the user know about the authentication result based on the comparison of the feature matrix of the newly collected data with the stored feature matrix of the legit user in the database.

3.4 Challenges

As this is an authentication-based system, the system should be able to validate the fine-grained signatures of the users using the wrist-worn device for PPG signal capture. In order to achieve that, there are primarily four challenges that needs to be addressed.

3.4.1 Coarse-grained Wrist PPG Signals

PPG signals are relatively more coarse-grained, noisier, and interfere with other signals than ECG signals. And wrist-worn PPG techniques are even more coarse-grained. As it can be seen from Fig. 3.5, the critical landmarks are more detectable

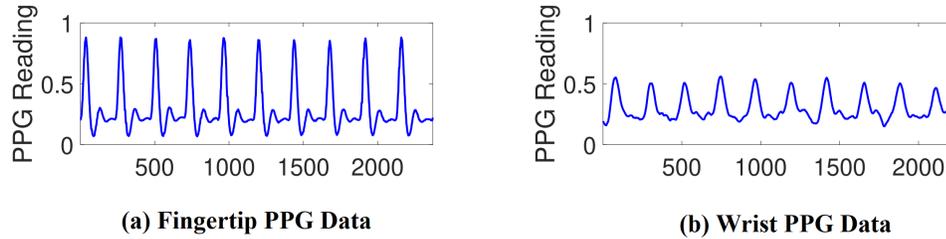


Figure 3.5: Example of PPG data from fingertip & wrist [1].

in the fingertip region based PPG signals while the landmarks in the wrist region are not. This means, methods or systems that are applicable to fingertip PPG techniques won't be applicable to wrist PPG technique. Besides these, the signal is generally contaminated with noises due to subtle hardware capture issues.

For handling this challenge, first we filtered the noise using different state of the art filters which are discussed in Section 3.5. And as the landmarks are less detectable and critical, we use various relevant feature extraction techniques to extract discriminating features which makes the authentication system a fine-grained one. The details of this feature extraction is explained in Section 3.6.

3.4.2 Different PPG Readings of Same User

This was one of the major challenges of the system initially when the plan was to use some of the strategies of the state of the art user authentication systems such as, [1], [64] where the user pulse is mainly considered for validating a user. In our system, we want to verify the users based on the signature or it can be said that we wanted to validate the signatures. If the whole signal is considered for validation which contains the pulse signal along with the signature signal of the user then the major issue is the variation of the PPG readings of the same user. As you can see in Fig. 3.6, the left sub-figure and the right sub-figure have a big variation in the sensor reading even though they represent the PPG sensor reading of the same user's pulse signal. The data is ranging between 430-610 in the left sub-figure and between 470-580 in the right sub-figure. This study brings out the fact that it is not feasible to use the pulse

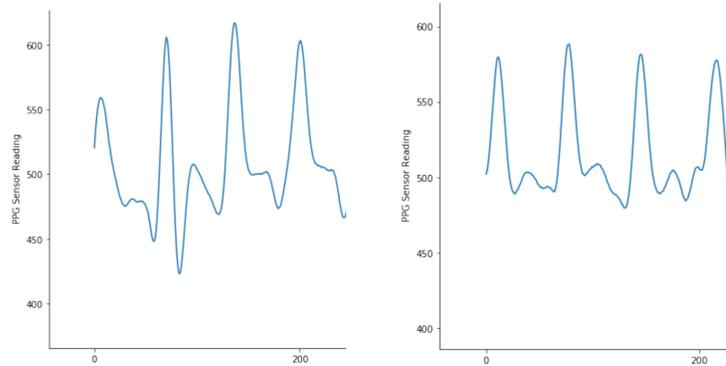


Figure 3.6: Example of a user with different PPG pulse signal.

signal along with the signature signal for the signature authentication as the pulse signal may vary due to the effect of pressure and emotions of the user.

As the same user has a different pulse signal in different situation, so we need to segment the signature portion and the pulse signal portion from the whole signal and remove the pulse signal portion so that the effect of pressure, and emotion is not a problem anymore. The details of the segmentation process is explained in Section 3.5.

3.4.3 Effect of PPG Sensor Placement

Another important challenge for the system is the effect of the placement of the PPG sensor. For our system, we built our own wrist-worn wearable device. For the custom built wrist-worn device, the sensor was not attached to the Velcro band so, we could change the position of the sensor according to our need. The details of the hardware will be explained in Section 4.1.

Generally, the pulse signal of a user is measured from the radial artery, the right red circle in the right sub-figure of Fig. 3.7. As the PPG sensor is generally used for measuring heart rate, blood pressure, and pulse oximetry, existing applications concentrate on the radial artery more. In our work, we focus on the minute movements of fingers along with the hand itself while writing the signature. PPG sensor readings vary if the placement of the sensor is effectively done. After a lot of iteration of the placement of the PPG sensor, we found out that if the sensor is placed around

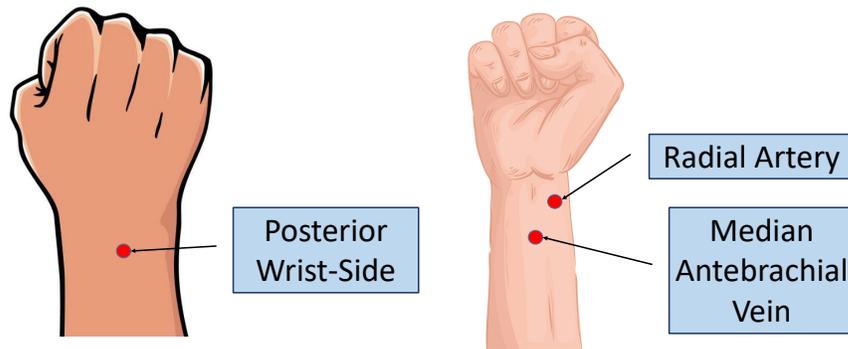


Figure 3.7: Location for placement of the PPG sensor.

the Median Antebrachial Vein (left red circle in the right sub-figure of Fig. 3.7), the readings would represent the significant motion artifacts occurring due to the signature writing. The commercially available wrist-worn device is generally worn with the sensor being placed on the posterior side of the wrist (shown in the leftmost red circle in Fig. 3.7) but for this work, we wear it on the anterior side of the wrist to cover the Median Antebrachial Vein to capture the minute motion artifacts caused by the signature writing.

3.4.4 Overlapped Signature Signals

The final challenge is the overlapping of pulse signals with signature signals in the overall received signal via the PPG sensor of the wrist-worn wearable device. Existing works do not consider the case of overlapping of signals, rather they focus on segmenting the motion artifacts part from the main signal [112, 1]. In our case we have to consider the motion artifacts only not the pulse signal. And even if we are able to segment the motion artifacts part i.e. the signature portion, there will be a mandatory overlapping with the pulse signal because the user's pulse is continuous. A schematic example is shown in Fig. 3.8.

The removal of overlapping of these two signals is still a challenge that needs

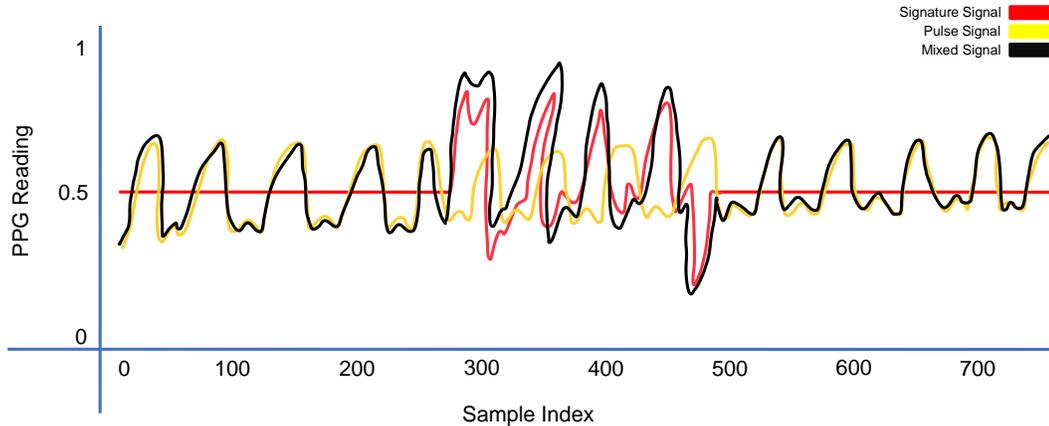


Figure 3.8: A schematic representation of POSA's signal overlapping.

to be further studied. In *PPGPass*[64], the authors want to remove the motion artifacts from the mixed signals to get the pure pulse signal for extracting intrinsic discriminating features of a user. This algorithm demands two sources of input in order to apply the blind source separation algorithms. One of our future work is to introduce a novel algorithm that can separate the mixed signals from only one source. Therefore, the signature separation part is not implemented in this dissertation.

3.5 Data Segmentation

In this section, we will see the details of the data segmentation module. As soon as the data is collected via the wrist-worn wearable device, it is being sent to the server for data processing and the first step that the data comes to, is the data segmentation module. We have divided the data segmentation flow into the following three phases:

- Pre-processing
- Signature Segmentation
- Signature Separation

The flow of the data segmentation module is briefly shown in Fig. 3.9.

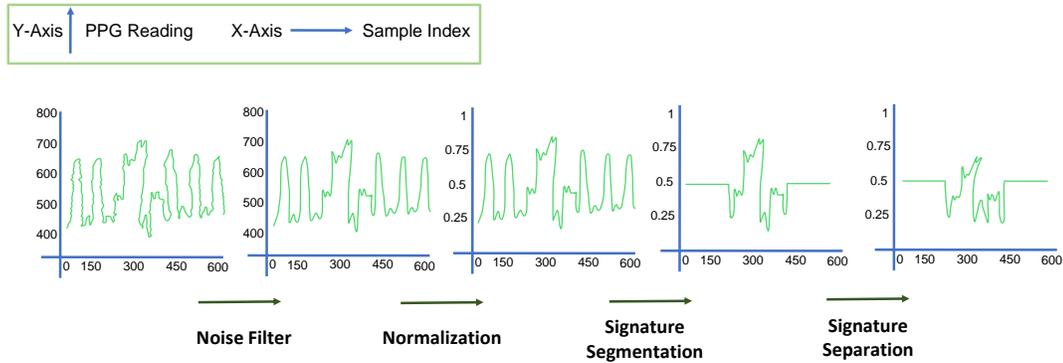


Figure 3.9: A schematic representation of data segmentation module flow.

3.5.1 Pre-processing

The first phase of the data segmentation module is the pre-processing step. After the PPG sensor data is collected using the wrist-worn device, the raw data is passed on to the server for further processing. Before starting the segmentation phase, the raw data needs to be pre-processed so as to be able to be worked on. The raw data comes into the system as ".csv" formatted files. The two steps for working on the data is *noise filtering* and *data normalization* which are discussed below.

Noise Filtering: Due to the user's behavioral changes and surrounding environmental changes, there are bound to be noises in the raw PPG data collected using the wrist-worn wearable devices. Also because of hardware imperfections, there are baseline drifts and high frequency interference in the PPG sensor readings [1].

The human heart rate is generally around 50-100 beats per minute [64]. That means the frequency is ranging from 0.8 - 1.7 Hz. Initially, we applied high band pass filter or Butterworth filter high cut-off with a frequency of 2 Hz. But we cannot actually use these filters because the signature portion might have also some element with low frequency which will get filtered and it should not be the case. So, for the initial pre-processing, we focus on the noises mentioned in the previous paragraph. To remove those noises, we apply a famous smoothing method called Savitzky-Golay (S-G) Filter [113]. Fig. 3.10 shows the effect of the application of the S-G filter.

Before applying the filter, you can see that the signal has some granular noises along the trajectory which is smoothed in the right sub-figure after applying the S-G filter. Savitzky-Golay filter can be called as a type of low-pass filter which is a smoothing method based on local least-squares polynomial approximation. The polynomial fitting of the sample points and the evaluation of it at a single point within the approximation is equivalent to discrete convolution with a fixed impulse response [114]. Savitzky and Golay were trying to smooth the data for Chemical spectrum analyzers. The technique of theirs using the least squares polynomial approximation have not only removed the noises but also was able to maintain the shape and height of the peaks of the signal's/sample's waveform.

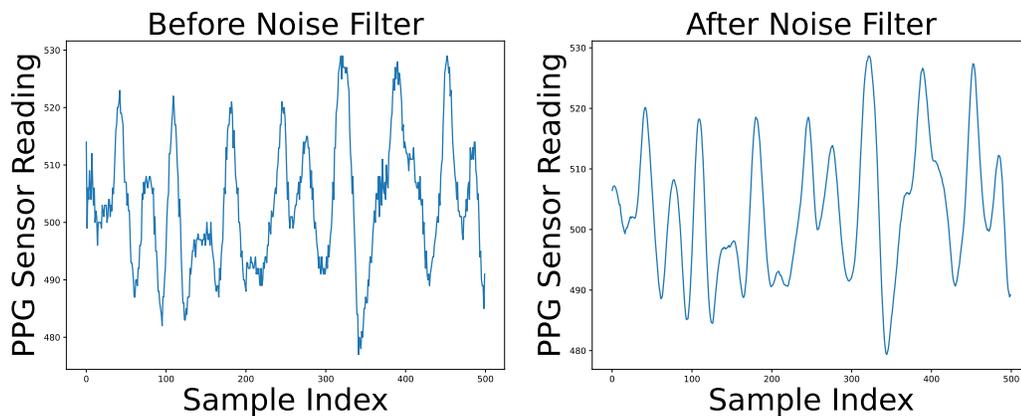


Figure 3.10: Comparison after applying noise filtering method.

Data Normalization: After the noises were removed in the noise filtering step, the noise filtered PPG data is passed on to the next step where the data is normalized. After a lot of experiments, we saw that the PPG data is generally ranging from 400-750. The range is quite big and different user have different variation of it within the range. It is a good practice in general to normalize a data before further processing it. The scale of data would be different resulting in a different scale of features for each different user in the feature characterization module in Section 3.6. Normalization

would handle this issue resulting in the training to be less of an problem to the scale of the data/features. After normalization, the whole dataset is within the range of 0 to 1 which would ultimately make the features more consistent with each other for a better training model. The effect of normalization is shown in Fig. 3.11 where the Y-axis of the left sub-figure ranges from 450 - 700 and the Y-axis of the right sub-figure is ranging from 0 to 0.03 which is within the range of 0-1. There are conditions for convergence of data which can be satisfied easily if normalization is done beforehand. To make an optimization problem work, the convergence problem can not have a big variance. And Normalization process helps to keep the variance less which leads to an optimized solution for an optimization problem. If the given signal is an array Y , and each data is y_i where i ranges from 0 to n (the length of Y), and the normalized data is y'_i then:

$$y'_i = \frac{y_i - \min(Y)}{\max(Y) - \min(Y)} \quad \text{where, } i = 0, 1, \dots, n.$$

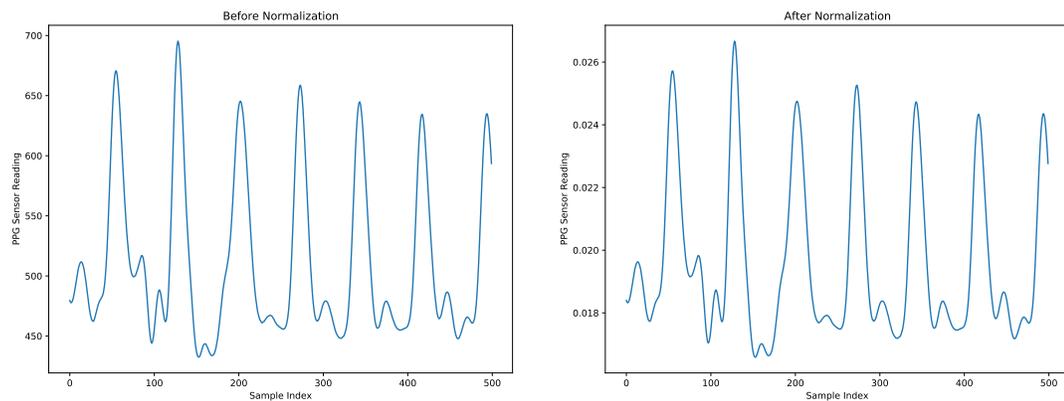


Figure 3.11: Comparison after applying data normalization method.

3.5.2 Signature Segmentation

The second and the most important phase of data segmentation is the step of signature segmentation. The raw data after being pre-processed i.e. after being noise

filtered and normalized is passed on to this step according to Fig. 3.9. In this step, the processed data is segmented in such a way that the signature portion of the data is obtained as an output. As we mentioned before that this data is a mixed signal of the pulse profile of the user and the signature motion of the user. So, even if we are able to segment the signature portion, the signal data for that portion will be a mixed signal.

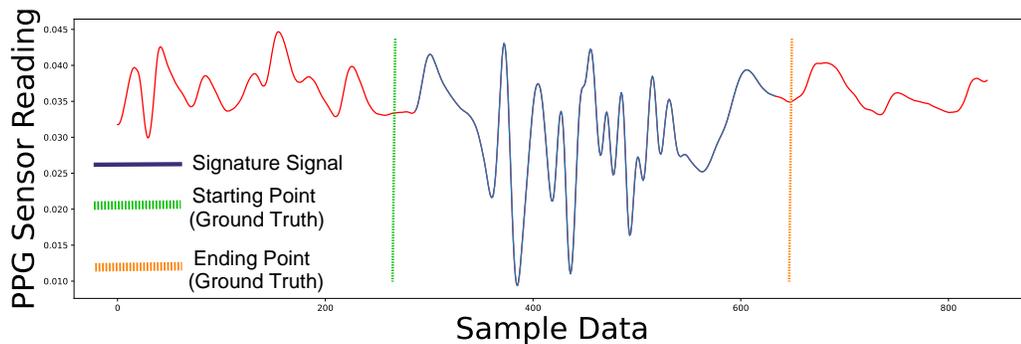


Figure 3.12: Segmentation of signature signal using Skewness-DTW method.

We have implemented multiple algorithms to segment the signature signals and run the whole framework of the system after the segmentation. The first algorithm that we designed is called the "*Skewness-DTW*" method. In this Algorithm 1, the input is the filtered & normalized signal of the user and the output is the start and end index of the signature portion of the signal. From Line 1 to 7 of the Algorithm 1, we find out the start index of the signature portion based on the skewness values of the signal with a fixed window size. After a lot of iterations, we found out that a window size of 100 is a good fit for our system. The index of the minimum value of skewness of the signal gives us the start point of the signature portion of the user. From Line 8 to 16 of the Algorithm 1, we find out the end index of the signature portion of the signal similar to [112]. This part of the Algorithm uses the Dynamic Time Warping technique to find out the end point. In the beginning, it requires a pulse profile of the user which is obtained from the first few seconds of the signal where the user didn't start writing his/her signature rather keeps their hand/wrist static. This

gives the system a portion of the signal which is just the pulse signal of the user. A sliding window based approach is taken where the window size is considered to be the average size of a single pulse of the user. In our case the pulse size was considered as 80. Each window of the signal was compared with the pulse profile of the user based on the dynamic time warping method and a score was generated which indicates the similarity between them. Finally, the end point was detected based on the minimum value of the scores obtained via the DTW method. Thus, we get both the start and end index of the signature portion of the signal using this "*Skewness-DTW*" method. Fig. 3.12 shows an example of the output.

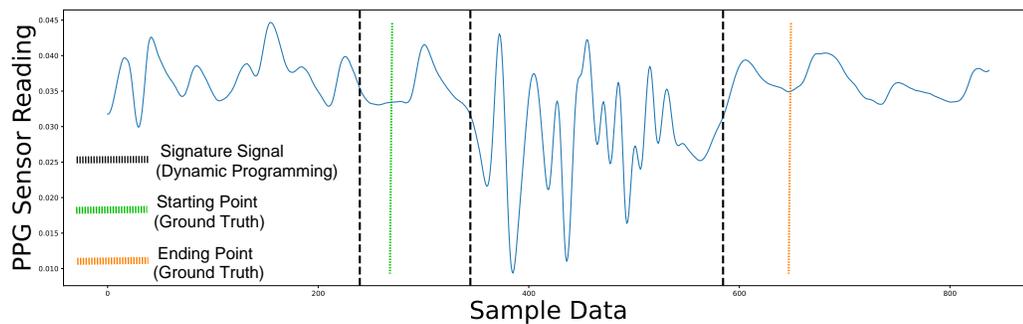


Figure 3.13: Segmentation of signature signal using dynamic programming method.

We implemented two more algorithms which are offline change point detection methods from [115]. The first algorithm is based on a dynamic programming and the second algorithm is based on a modified binary search method. The methods are shown in Algorithm 2 and Algorithm 3. In the Algorithm 2, the costs of all the subsequences of the signal is computed at first and then the minimum of the sum of the costs is calculated. In this process, the number of change points to detect has to be predefined. After some iterations, we found out that when the number of change points is 3, the accuracy of the segmentation is the best where the first and last change points are the start & end points of the signature signal. Fig. 3.13 shows an example of the output. The complexity of the Algorithm 2 is $\mathcal{O}(CKn^2)$, where C is the complexity of calling the considered cost function on one sub-signal, K is

the predefined number of change points, and n is the number of samples of the given signal.

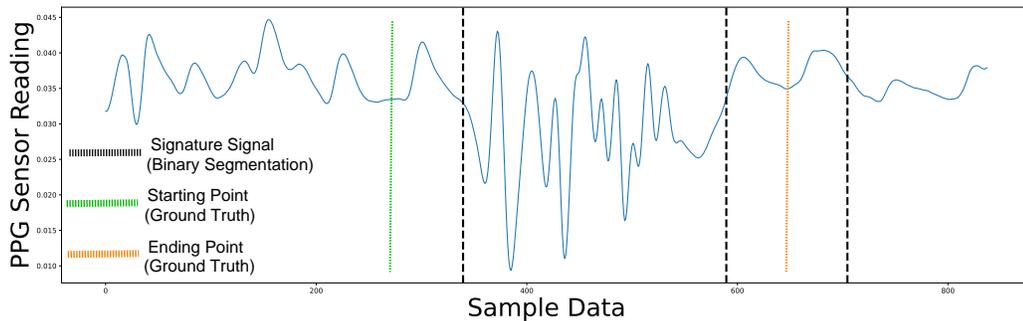


Figure 3.14: Segmentation of signature signal using binary segmentation method.

Algorithm 3 is a greedy sequential algorithm. It is greedy because it searches for the point for which the sum of the costs gets lowered. After the first change point is detected, the signal is split into two parts on that point. The algorithm is repeated on the sub-signals until the stopping condition is satisfied. As it relates to the process of binary search, it is called the Binary Segmentation method. The complexity of the Algorithm 3 is $\mathcal{O}(Cn \log n)$, where C is the complexity of calling the considered cost function on one sub-signal, and n is the number of samples of the given signal. For this algorithm, it is fine even if there are no predefined number of change points. But after a lot of iterations, we found that the system gives the best output when the number is predefined as 3 where the first and the last change points are the start & end points of the signature signal. Fig. 3.14 shows an example of the output.

Another possible segmentation technique we intended to do is a design-based segmentation method. This design-based method would broadcast a beep sound from the wrist-worn wearable device to indicate the start time & end time of the writing of the signature. By using this technique, we restrict the users to sign with a certain time range that we have defined using the design of the beep sound. Now, as we already know in which start time and end time the signature is in within the signal, we can easily segment the signature portion from the main signal.

Algorithm 1 Skewness-DTW Method

Input: Filtered & Normalized Signal $S(n)$

Output: Start and End index of Signature Portion of the signal

```

1:  $windowSize = 100$ 
2: DECLARE  $skewList$ : LIST
3: for  $i = 0 \rightarrow (length\ of\ S(n) - windowSize)$  do
4:    $value = SKEWNESS(S[i : (i + windowSize)])$ 
5:   Append  $value$  to  $skewList$ 
6: end for
7:  $startPoint = IndexOfMin(skewList)$ 
8:  $pulseSize = 80$ 
9: DECLARE  $scoreList$ : LIST
10: for  $i = 0 \rightarrow (length\ of\ S(n) - pulseSize)$  do
11:    $pulseProfile = S[: pulseSize]$ 
12:    $dataCompare = S[i : (i + pulseSize)]$ 
13:    $score = DTW(pulseProfile, dataCompare)$ 
14:   Append  $score$  to  $scoreList$ 
15: end for
16:  $endPoint = IndexOfMin(scoreList)$ 
17: return  $startPoint, endPoint$ 

```

Algorithm 2 Dynamic Programming Method

Input: Filtered & Normalized Signal $S(n)$

Output: Start and End index of Signature Portion of the signal

```

1: Assign  $linearl_1$  to Cost Function
2: Fitting the signal based on the Cost Function
3:  $bkps = Calling\ Prediction\ function\ for\ breakpoints$ 
4:  $startPoint = bkps[0]$ 
5:  $endPoint = bkps[-2]$ 
6: return  $startPoint, endPoint$ 

```

3.5.3 Signature Separation

This subsection deals with the challenge mentioned earlier in Section 3.4.4. The challenge was the overlapping of pulse signals with signature signals in the overall received signal via the PPG sensor of the wrist-worn wearable device. The state of the art systems do not consider the case of overlapping signals [112, 1]. Rather they concentrate on the segmentation part only. In our case, the motion artifact portion is the signature portion which is the main concern. Cao *et al.* in [64], even though they

Algorithm 3 Binary Segmentation Method

Input: Filtered & Normalized Signal $S(n)$

Output: Start and End index of Signature Portion of the signal

- 1: Assign *normal* to Cost Function
 - 2: Fitting the signal based on the Cost Function
 - 3: $bkps = \text{Calling Prediction function for breakpoints}$
 - 4: $startPoint = bkps[0]$
 - 5: $endPoint = bkps[-2]$
 - 6: **return** $startPoint, endPoint$
-

considered the overlapping of the signals, their concern was to get the pulse profile of the user for authentication. But for our system, even if we are able to segment the motion artifacts part i.e. the signature portion, there will be a mandatory overlapping with the pulse signal because the user's pulse is continuous. A schematic example was shown in Fig. 3.8 before. The algorithm mentioned in [64] demands two sources of input in order to apply a blind source separation algorithm. They have applied a two stage motion artifact removal algorithm where the first stage uses a modified semi-blind source separation (S-BSS) algorithm [116] to estimate the pulse signals and the motion artifacts assuming the fact that they are linearly combined with each other. In [64], green and infrared light data were used as the two sources for the S-BSS algorithm. In the first stage of the algorithm, it was assumed that the signals are linearly mixed but in reality they are not ideally linearly mixed for which Cao *et al.* implemented an adaptive filter known as adaptive step-size least mean squares (AS-LMS) to remove the motion artifacts [117].

To adopt this two stage algorithm we would need two sources of the signal. But instead of removing the motion artifacts, in our case which is the signature portion, we would store the signature portion as it is the main focus of our scope. Our developed hardware prototype for collecting the PPG data from the user is compromised of only a single PPG sensor. Thus the two-stage algorithm from [64] can not be applied in this case. One of our future work is to introduce a novel algorithm that can separate the mixed signals from only one source. For this dissertation, the signature separation

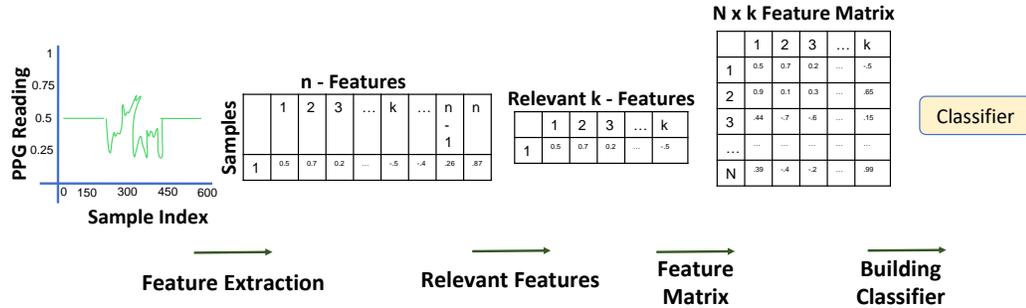


Figure 3.15: Feature characterization module flow.

step was not implemented. Even without the signature separation step, our system had satisfactory performance discussed in Section 4.3.

3.6 Feature Characterization

After the collected raw PPG data is being processed in the data segmentation module and a noise filtered, normalized, and segmented signature signal is received, the signature portion of the data is passed on to the Feature Characterization module. The Feature Characterization module is divided into four phases:

- Feature Extraction
- Relevant Features
- Feature Matrix
- Classifiers

In this subsection, various PPG features are explored in order to facilitate the signature authentication or one can say signature recognition for a certain user. Then those features are built into a feature matrix which helps to build the classifier required to classify the signature into either legitimate user or an attacker class. The flow of the Feature Characterization module is briefly shown in Fig. 3.15.

3.6.1 Feature Extraction

As we have the clean signature portion now, the feature characterizing process can be started. The first step of the feature Characterization module is the feature extraction of the signature PPG data. The PPG features that we focus on are the time-domain features. For feature extraction, we used a software tool called *TSFRESH* [118]. *TSFRESH* is a python package that is used to automatically calculate a huge number of time series characteristics or features. By using about 63 characterization methods, it computes more than 700 time series features in an accelerated way compared to existing time-consuming processes. The clean and segmented signature data is converted into suitable dataframe which could be processed by the methods of *tsfresh* package. The features that *tsfresh* can extract are listed in Tables 3.1-3.4 [119].

3.6.2 Relevant Features

The second phase of the feature Characterization module is the relevant feature selection. This means the proper selection of strong and weak features which is a hard problem for a time-series classification. *tsfresh* incorporates a feature extraction process based on scalable hypothesis tests. It is really efficient, as it filters the important features to be used for the machine learning model beforehand which helps to train the model well as there are less percentage of irrelevant features.

The feature vectors that were generated in the previous step are all individually tested to predict the labels in regards of their significance. The output of these tests give a score vector for each feature vector. Later, these score vectors are evaluated on the basis of the Benjamini-Yekutieli procedure to decide which features to be discarded and which features are important enough to be kept[120].

After running the tool, we found **763 features** out of which **36 features** were selected as relevant features by *TSFRESH*. The selected features can be summarized

Table 3.1: List of Extracted Features by *tsfresh*.

Features	Description
$\text{abs_energy}(x)$	Returns the absolute energy of the time series which is the sum over the squared values.
$\text{absolute_sum_of_changes}(x)$	Returns the sum over the absolute value of consecutive changes in the series x .
$\text{agg_autocorrelation}(x, \text{param})$	Calculates the value of an aggregation function f_{agg} .
$\text{agg_linear_trend}(x, \text{param})$	Calculates a linear least-squares regression for values of the time series that were aggregated over chunks versus the sequence from 0 up to the number of chunks minus one.
$\text{approximate_entropy}(x, m, r)$	Implements a vectorized Approximate entropy algorithm.
$\text{ar_coefficient}(x, \text{param})$	This feature calculator fits the unconditional maximum likelihood of an autoregressive $AR(k)$ process.
$\text{augmented_dickey_fuller}(x, \text{param})$	The Augmented Dickey-Fuller test is a hypothesis test which checks whether a unit root is present in a time series sample.
$\text{autocorrelation}(x, \text{lag})$	Calculates the autocorrelation of the specified lag.
$\text{benford_correlation}(x)$	Useful for anomaly detection applications.
$\text{binned_entropy}(x, \text{max_bins})$	First bins the values of x into max_bins equidistant bins.
$\text{change_quantiles}(x, ql, qh, isabs, f_agg)$	First fixes a corridor given by the quantiles ql and qh of the distribution of x .
$\text{cid_ce}(x, \text{normalize})$	This function calculator is an estimate for a time series complexity (A more complex time series has more peaks, valleys etc.).
$\text{count_above}(x, t)$	Returns the percentage of values in x that are higher than t .
$\text{count_above_mean}(x)$	Returns the number of values in x that are higher than the mean of x .
$\text{count_below}(x, t)$	Returns the percentage of values in x that are lower than t .
$\text{count_below_mean}(x)$	Returns the number of values in x that are lower than the mean of x .
$\text{cwt_coefficients}(x, \text{param})$	Calculates a Continuous wavelet transform for the Ricker wavelet, also known as the ‘‘Mexican hat wavelet’’.

Table 3.2: Continued List of Extracted Features by *tsfresh*.

Features	Description
<code>energy_ratio_by_chunks(x, $param$)</code>	Calculates the sum of squares of chunk i out of N chunks expressed as a ratio with the sum of squares over the whole series.
<code>fft_aggregated(x, $param$)</code>	Returns the spectral centroid (mean), variance, skew, and kurtosis of the absolute fourier transform spectrum.
<code>fft_coefficient(x, $param$)</code>	Calculates the fourier coefficients of the one-dimensional discrete Fourier Transform for real input by fast.
<code>first_location_of_maximum(x)</code>	Returns the first location of the maximum value of x .
<code>first_location_of_minimum(x)</code>	Returns the first location of the minimal value of x .
<code>fourier_entropy(x, $bins$)</code>	Calculate the binned entropy of the power spectral density of the time series (using the welch method).
<code>friedrich_coefficients(x, $param$)</code>	Coefficients of polynomial $h(x)$.
<code>has_duplicate(x)</code>	Checks if any value in x occurs more than once.
<code>has_duplicate_max(x)</code>	Checks if the maximum value of x is observed more than once.
<code>has_duplicate_min(x)</code>	Checks if the minimal value of x is observed more than once.
<code>index_mass_quantile(x, $param$)</code>	Those apply features calculate the relative index i where $q\%$ of the mass of the time series x lie left of i .
<code>kurtosis(x)</code>	Returns the kurtosis of x (calculated with the adjusted Fisher-Pearson standardized moment coefficient G2).
<code>large_standard_deviation(x, r)</code>	Boolean variable denoting if the standard dev of x is higher than 'r' times the range = difference between max and min of x .
<code>last_location_of_maximum(x)</code>	Returns the relative last location of the maximum value of x .
<code>last_location_of_minimum(x)</code>	Returns the last location of the minimal value of x .
<code>lempel_ziv_complexity(x, $bins$)</code>	Calculate a complexity estimate based on the Lempel-Ziv compression algorithm.
<code>length(x)</code>	Returns the length of x .
<code>linear_trend(x, $param$)</code>	Calculate a linear least-squares regression for the values of the time series versus the sequence from 0 to length of the time series minus one.
<code>linear_trend_timewise(x, $param$)</code>	Calculate a linear least-squares regression for the values of the time series versus the sequence from 0 to length of the time series minus one.
<code>longest_strike_above_mean(x)</code>	Returns the length of the longest consecutive subsequence in x that is bigger than the mean of x .

Table 3.3: Continued List of Extracted Features by *tsfresh*.

Features	Description
<code>longest_strike_below_mean(x)</code>	Returns the length of the longest consecutive subsequence in x that is smaller than the mean of x .
<code>max_langevin_fixed_point(x, r, m)</code>	Largest fixed point of dynamics $:\text{math:argmax}_x h(x)=0'$ estimated from polynomial $h(x)$.
<code>maximum(x)</code>	Calculates the highest value of the time series x .
<code>mean(x)</code>	Returns the mean of x .
<code>mean_abs_change(x)</code>	Returns the mean over the absolute differences between subsequent time series values.
<code>mean_change(x)</code>	Returns the mean over the differences between subsequent time series values.
<code>mean_second_derivative_central(x)</code>	Returns the mean value of a central approximation of the second derivative.
<code>median(x)</code>	Returns the median of x .
<code>minimum(x)</code>	Calculates the lowest value of the time series x .
<code>number_crossing_m(x, m)</code>	Calculates the number of crossings of x on m .
<code>number_cwt_peaks(x, n)</code>	This feature calculator searches for different peaks in x .
<code>number_peaks(x, n)</code>	Calculates the number of peaks of at least support n in the time series x .
<code>partial_autocorrelation(x, param)</code>	Calculates the value of the partial autocorrelation function at the given lag.
<code>percentage_of_reoccurring_datapoints_to_all_datapoints(x)</code>	Returns the percentage of non-unique data points.
<code>percentage_of_reoccurring_values_to_all_values(x)</code>	Returns the percentage of values that are present in the time series more than once.
<code>permutation_entropy(x, tau, dimension)</code>	Calculate the permutation entropy.
<code>quantile(x, q)</code>	Calculates the q quantile of x .
<code>range_count(x, min, max)</code>	Count observed values within the interval $[min, max)$.
<code>ratio_beyond_r_sigma(x, r)</code>	Ratio of values that are more than $r * std(x)$ (so r sigma) away from the mean of x .
<code>ratio_value_number_to_time_series_length(x)</code>	Returns a factor which is 1 if all values in the time series occur only once, and below one if this is not the case.
<code>sample_entropy(x)</code>	Calculate and return sample entropy of x .

Table 3.4: Continued List of Extracted Features by *tsfresh*.

Features	Description
<code>set_property(key, value)</code>	This method returns a decorator that sets the property <i>key</i> of the function to <i>value</i> .
<code>skewness(x)</code>	Returns the sample skewness of x (calculated with the adjusted Fisher-Pearson standardized moment coefficient G1).
<code>spkt_welch_density(x, param)</code>	This feature calculator estimates the cross power spectral density of the time series x at different frequencies.
<code>standard_deviation(x)</code>	Returns the standard deviation of x .
<code>sum_of_reoccurring_data_points(x)</code>	Returns the sum of all data points, that are present in the time series more than once.
<code>sum_of_reoccurring_values(x)</code>	Returns the sum of all values, that are present in the time series more than once.
<code>sum_values(x)</code>	Calculates the sum over the time series values.
<code>symmetry_looking(x, param)</code>	Boolean variable denoting if the distribution of x looks symmetric.
<code>value_count(x, value)</code>	Count occurrences of value in time series x .
<code>variance(x)</code>	Returns the variance of x .
<code>variance_larger_than_standard_deviation(x)</code>	Boolean variable denoting if the variance of x is greater than its standard deviation.
<code>variation_coefficient(x)</code>	Returns the variation coefficient (standard error / mean, give relative value of variation around mean) of x .

Table 3.5: List of Relevant Features selected by *tsfresh*.

Features	Description
<code>count_below_mean(x)</code>	Returns the number of values in x that are lower than the mean of x .
<code>number_peaks(x, n)</code>	Calculates the number of peaks of at least support n in the time series x .
<code>range_count(x, min, max)</code>	Count observed values within the interval $[min, max)$.
<code>fft_coefficient(x, param)</code>	Calculates the fourier coefficients of the one-dimensional discrete Fourier Transform for real input by fast.
<code>number_cwt_peaks(x, n)</code>	This feature calculator searches for different peaks in x .
<code>absolute_sum_of_changes(x)</code>	Returns the sum over the absolute value of consecutive changes in the series x .
<code>agg_linear_trend(x, param)</code>	Calculates a linear least-squares regression for values of the time series that were aggregated over chunks versus the sequence from 0 up to the number of chunks minus one.
<code>linear_trend(x, param)</code>	Calculate a linear least-squares regression for the values of the time series versus the sequence from 0 to length of the time series minus one.

into primarily **8 features** which are shown in Table 3.5.

3.6.3 Feature Matrix

Now that we have all the extracted features and selected relevant features of the user's signature signal, we work on the third phase of the feature characterization module which is feature matrix. After collecting n calibrating sample signals from a user, the relevant 36 features are generated for each sample. The next step is feature scaling which is one of the most important step before passing the data onto training a classifier. Different machine learning algorithms are insensitive to the scale of diversified features. Each feature value might have a different range either a higher value or a lower value. But if feature scaling is done then the convergence for the machine learning algorithms is faster. In our system, for feature scaling, we use standardization of the data. Say, the new value for a certain feature is Y_{new} then:

$$Y_{new} = \frac{Y - mean}{Standard\ Deviation}.$$

After standardizing the features, we form a $2D$ matrix of the dimension $n * 36$ where each row represents the signature sample data and each column represents a selected relevant feature.

3.6.4 Classifiers

The final and the most important step of Feature Characterization module is building and training the classifiers. So far we have collected the raw PPG data then noise filtered it, normalized it, segmented the signature portion, extracted the features, selected the relevant features, standardized the features, and generated a feature matrix. Now we train the model based on this feature matrix to build a classifier. For our classifiers, we tested on the following standard classifiers:

- Random Forest (RF)
- Support Vector Machine (RBF Kernel)
- Gradient Boosting (GB)
- k-nearest neighbor (kNN)
- Multilayer Perceptron (MLP)
- Feed-Forward Neural Network (NN)
- OnevsRest (OvR)

These classifiers will determine whether the new input signature PPG data belongs to a legitimate user or an attacker.

CHAPTER 4: POSA IMPLEMENTATION AND EVALUATION

In this chapter, we discuss the implementation of the so far discussed modules of the POSA system and the subsequent evaluation of the system’s different pre-mentioned classifiers based on the experiments.

4.1 Experimental Setup

This subsection explains the implementation of the experimental setup that we built for our experiments. The experimental setup has two folds: POSA Band and Server. The POSA Band is built from the scratch from commercially off-the shelf components and the server is set up on a commercially off-the shelf laptop. The details of these setups are further discussed below in Section 4.1.1 and Section 4.1.2.

4.1.1 POSA Band

The commercially available smart watches and fitness trackers though they use PPG sensors to measure the heartbeat/pulse, most of them do not provide access to the raw PPG data. For this reason, the first device that we use for our experiments is a low-cost proof-of-concept prototype, designed and implemented by us and we named it as “*POSA Band*”. This POSA Band would try to imitate the wrist-worn wearable devices to validate the feasibility of POSA. The different components of POSA Band is shown in Fig. 4.1. The prototype that we developed consists of a velcro wristband, a PPG sensor with a green LED (as green LED performs the best), a USB cable (to connect the micro-controller to the server), an Arduino UNO micro-controller, and a slide switch to start and stop collecting data.

The PPG sensor is from “World Famous Electronics llc.” and has three wires connected to it which are red, black, and violet. The red wire has to be connected

to the 5V port of the Arduino and the black & violet wire have to be connected to the Ground port and Pin A0 of the Arduino. For the slide switch to work it was connected to the 3.3V port and the other ends were connected to the GND and pin 12 of the micro-controller. The USB cable is connected as the communication between the Arduino and the server which is a Dell Inspiron 15 laptop in our case. The pulse sensor is strapped to the Velcro band so that it remains facing towards the wrist when the Velcro band is worn on the wrist to imitate the commercially available wrist-worn wearable.

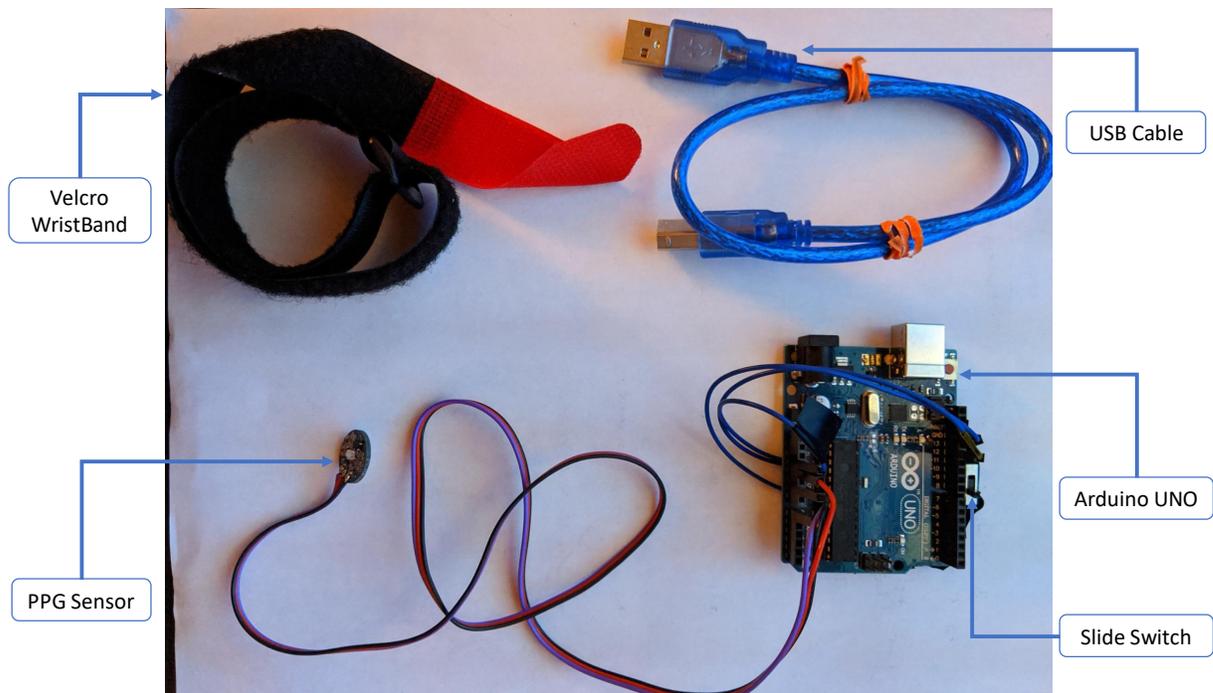


Figure 4.1: Overall hardware setup for data collection via POSA band.

4.1.2 Server

In our system, we used a Dell Inspiron 15 7000 laptop as our server which has a 16 GB RAM, Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz, 2801 Mhz, 4 Core(s), 8 Logical Processor(s), & a NVIDIA GeForce GTX 1050 Ti GPU.

The processing modules of POSA are all programmed into the server in *Python* 3.8.3. Algorithm 2 and Algorithm 3 were implemented using the library package called “rup-

tures” mentioned in [115]. For feature extraction, we used a package called “tsfresh” mentioned in [118]. For building the different types of classifiers, we used the python package *scikit-learn* [121].

4.2 Data Collection

In this section, we discuss how the POSA Band (Section 4.1.1) is used to collect the user’s raw PPG data while writing the signature.

The user is seating on a chair resting his/her hand on a piece of paper on which he/she will provide the signature with a pen or it can also be the case that the user is resting the hand on a smart device screen on which he/she will provide the signature with a digital pen. The POSA band is wrapped around the wrist of the user with the PPG sensor facing towards the Median Antebrachial vein discussed in Section 3.4.3. The data collection setup is shown in Fig. 4.2.

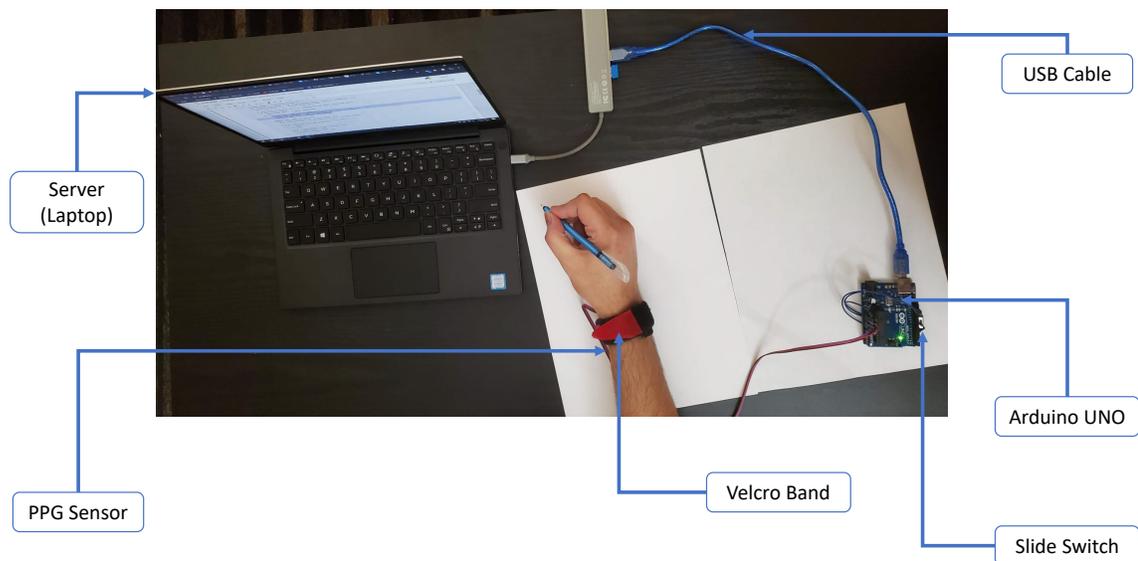


Figure 4.2: Data collection using the POSA band prototype.

After the slide switch is pressed, the raw stream of PPG data at a sample rate of **97.5 Hz** starts flowing to the server from the PPG sensor via the Arduino microcontroller. In the server side, a software called “**TeraTerm**”, which is a computer

terminal program and supports Serial Port connections, is used. TeraTerm captures the raw PPG data coming via the serial port connection and saves it into a “.csv” formatted file. This “.csv” formatted file is then passed on to the advanced signal processing and artificial intelligent algorithms implemented in *Python* 3.8.3.

For this dissertation, we designed the experiment with pen and paper-based signatures and did not include the digital signatures or online signatures. As our system relies on the minute movements of the user’s fingers and hand movement during a signature writing, logically, it should have the same performance for both offline and online signature authentication. Because of the logistic limitations being created due to the COVID-19 pandemic situation at the time, 5 healthy volunteers participated in the experiment. Participants take part in 6 sessions of signature writing. During each session, the participants provide 20 valid signatures of themselves, and 10 random forgeries and 10 skilled forgeries against each of the other four users. We adopt the same definition of random and skilled forgeries from [107]. In random forgeries, the attacker does not know the legitimate user’s signature, while in skilled forgeries the attacker trains on the claimed user’s signature.

4.3 Evaluation

Extensive performance evaluation has been conducted on our POSA system. For the evaluation, we have tested the data collected via *POSA Band* on multiple classifiers with the different splitting of training and test data. In our work, we adopted the user-specific model training where a model is trained for each user. In the evaluation, random forgery and skilled forgery attacks on the legitimate user from other users were primarily focused upon. In addition to these evaluations, we also tested a special case of attack where the legitimate user fakes his signature. We perform tests on three signature segmentation algorithms from Section 3.5 and all classifiers mentioned in Section 3.6.4. For splitting training and test data, we use a different portion of samples as the training data, from 20% up to 80%.

4.3.1 Metrics

For the quantitative analysis of the system, three main metrics were selected which are mentioned below:

Precision: Precision is the ratio of correctly predicted positive values to the total predicted positive values.

$$Precision = \frac{TP}{TP + FP}.$$

Recall: Recall is the ratio of correctly predicted positives values to the actual positive values.

$$Recall = \frac{TP}{TP + FN}.$$

F1 score: F1 score is a ratio of the positive and negative class. The formula for F1 score is as follows:

$$F1 = \frac{2 * Precision * Recall}{(Precision + Recall)}.$$

Here, $TP = True\ Positive$, $FP = False\ Positive$, and $FN = False\ Negative$.

For our system, high recall and low precision values mean that the positive cases i.e. the legitimate users are recognized but there are a lot of false positives i.e. approves illegitimate users. On the other hand, low recall and high precision values mean that the system has low false positives but high false negatives.

4.3.2 Performance

We have conducted performance evaluation on our current POSA system. For the evaluation, we tested the data collected via **POSA Band** on multiple classifiers with different splitting of training and test data. The classifiers that were used are Random Forest (RF), Support Vector Machine (SVM) (RBF Kernel), Gradient Boosting (GB), k-nearest neighbor (kNN), Multilayer Perceptron (MLP), Feed-Forward Neural Network (NN), and OnevsRest (OvR). Three types of segmentation techniques were used which are Algorithm 1, Algorithm 2, and Algorithm 3. Next we report the

detailed experimental results based on the impacts of different factors, such as types of segmentation methods, classifiers, training sizes, placement of the sensor, attack from same user, and different surfaces. All the results are based on the random and skilled forgery attacks except "attack from same user".

Impact of Segmentation Methods: In Section 3.5, three segmentation algorithms (Algorithm 1, Algorithm 2, and Algorithm 3) were introduced. The average performance of POSA for the segmentation methods with all the classifiers when the dataset was split into 60% training and 40% testing is shown in Fig. 4.3. Skewness-DTW and Dynamic Programming-based segmentation methods achieves an F1 score of 92% and 93% respectively. So, we can say that these two methods both performed well. In the rest of our evaluations, the results where the Skewness-DTW method is used, is reported.

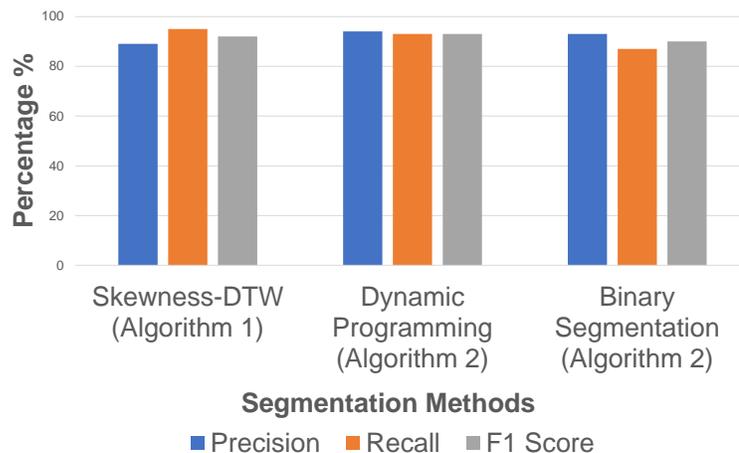


Figure 4.3: Performance of POSA for different segmentation algorithms averaged over all classifiers with 60% training data.

Impact of Training Sizes: In our experiment, for the training size of the data, we chose different percentages of the data. The average performance of Skewness-DTW segmentation method and all the classifiers when training data size is from 20% to 80% is shown in Fig. 4.4. The average F1-score increases from 86% to 94% overall when the size of the training dataset is increased. This also proves the fact that our

system can achieve a reasonable performance even with a lower training size.

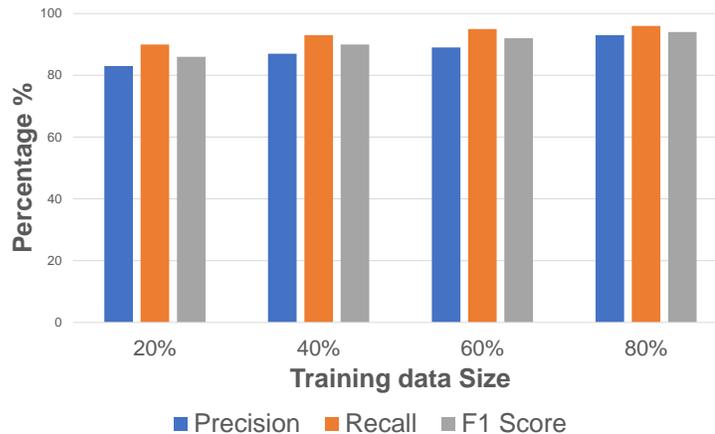


Figure 4.4: Performance of POSA for different training size varying from 20%-80% averaged over all classifiers.

Impact of Classifiers: In our system, we compare 7 commonly used classifiers: Random Forest (RF), Support Vector Machine (RBF Kernel), Gradient Boosting (GB), k-nearest neighbor (kNN), Multilayer Perceptron (MLP), Feed-Forward Neural Network (NN), and OnevsRest (OvR). The classifiers are considered under Skewness-DTW segmentation and with a training size of 80%. The F1-scores for each classifier are shown in Fig. 4.5. According to Fig. 4.5, Feed-forward Neural Network (NN) performed the best with an F1 score of 98% while Random Forest (RF), Gradient Boosting (GB) and OnevsRest (OvR) perform well consistently. The average performance for RF, GB and OvR over all the segmentation methods are 95%, 96%, and 95% respectively. As of now, multiple classifiers are performing well and one of our future work is to incorporate ensemble learning where these multiple classifiers will be combined conceptually. Then each classifier will return a label for the given data and the label that gets the most votes from the classifiers would be returned by the ensemble classifier as the final predicted label. As multiple models are used for ensemble learning, the probability of getting better predictive performance increases.

Impact of Placement of Sensor: As we mentioned earlier in Section 3.4.3,

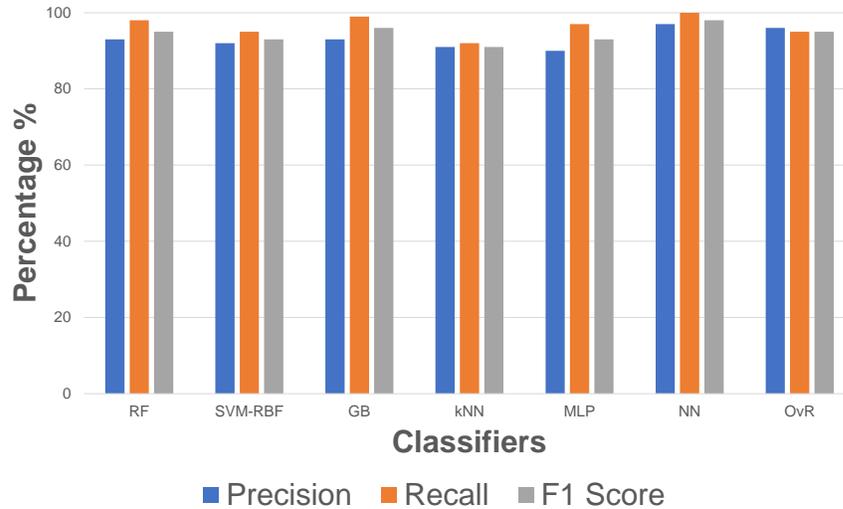


Figure 4.5: Performance of POSA for different classifiers (Skewness-DTW segmentation with 80% training data).

one of the challenges of the system was the placement of the sensor within the wrist area. In general, for pulse measurement PPG sensors are placed near the radial artery which is conventionally considered for measuring heart rate, blood pressure, and pulse oximetry. But for minute finger movements, the change in blood flow in Median Antebrachial Vein (left red circle in the right sub-figure of Fig. 3.7) region is most likely to occur. Also, in smartwatches/fitness trackers the sensors are beneath the watches and it is worn on the posterior side of the wrist. We also placed the sensor on that side of the wrist to see the feasibility for our system. A preliminary result of the comparison of the signature portion signal for different placement of the PPG sensor is shown in Fig. 4.6. As you can see, even though the same user wrote the same signature, for different location the signature portion is different. It is really hard to distinguish the signature portion from the pulse signal in case of Radial Artery and Posterior Wrist-side data. But when the placement of the PPG sensor is around the Median Antebrachial Vein, the signature portion is very distinguishable among the pulse signals. This proves the feasibility of the placement of the PPG sensor around the Median Antebrachial Vein region for our system. Such placement is also feasible

since the user can rotate the smartwatch/tracker 180° so that its PPG sensor faces the anterior side of the wrist when writing the signature.

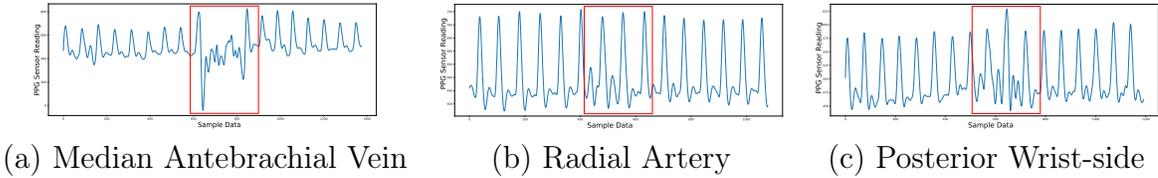


Figure 4.6: Comparison of the signature portion of the three different locations of PPG-sensor placement (a) median antebrachial vein (b) radial artery (c) posterior wrist-side.

Impact of Surface: The matter of surface comes into place for the offline signature scenario where the user is signing on a document placed on a certain surface. For online scenario, conventionally the option is the same where you need to sign using a digital pen on either a smart device screen (Resistive or Capacitive) or a digital screen. In case of offline signatures, the signing happens on a piece of paper, though the surface the paper is placed on might be different such as, plastic, wood, granite, and metal. For evaluating the impact of different surfaces, we collected 30 new samples for each type of surfaces . A preliminary result of the performance of the system for different surfaces based on Skewness-DTW segmentation method are shown in Fig. 4.7. As we can see from Fig. 4.7, the performance of the POSA system is almost the same in each of the scenarios which means there is no impact of surfaces on the system unless the surface is really uneven for which the finger, and wrist movements are irregular.

Attack from Same User: This is a unique type of attack where the user himself is trying to falsify signing the document in the system by signing other than his signature. For example, a user is legitimately ordered to sign a document which he/she does not want to sign. So the user tries to forge his own signature to escape from the situation. Our system shows a promising performance to handle this type of

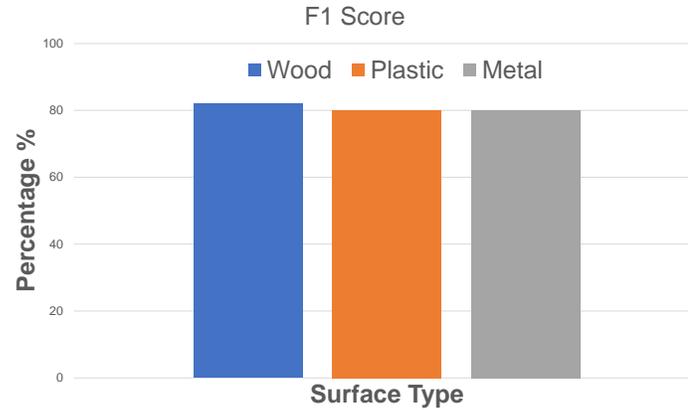


Figure 4.7: Performance of POSA on the three different surfaces (a) wood (b) metal (c) plastic.

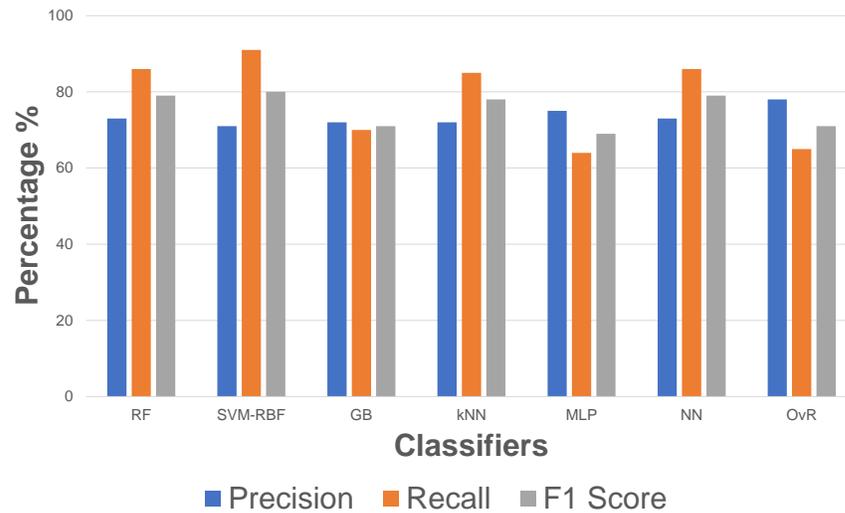


Figure 4.8: Performance of POSA under attacks from the same user.

attack. Here, for each user's dataset, only his invalid signatures are used as the forged signatures. The result is shown in Fig. 4.8, with near 80% scores for 20% training data.

CHAPTER 5: PPG-BASED WEIGHT LIFTING ASSESSMENT (PaWLA)

5.1 Introduction

An important key to a healthy life is Physical Activity (PA) which reduces the possibility of having different kinds of chronic disease such as, cardiovascular diseases, hypertension, obesity, depression, diabetes, muscular tissue damages, and respiratory illness. Among the PAs, stationary exercises such as, strength training is an important part of routine workout sessions [122]. Another important thing is to be aware of the intensity of the performed exercise to receive the full advantage of the training. Many research and investigation has been done in both industry and academia to recognize the intensity of stationary exercises like strength training. Different systems used different modality to serve the purpose like video, garment, and wearable sensor [123, 124, 70].

Besides physical activities, there are different scenarios where a person needs/wants to be aware of the quantitative intensity of the strength activity like weightlifting. For example, when a woman is pregnant she does not want be in such a situation where she is lifting a highly weighted object because she is not aware about the quantitative intensity of the physical act. Another scenario can be for a daily labor working in a construction site. For his daily work, he might need to lift weighted objects but there is obviously a certain amount of weight that he should never exceed to avoid suffering from muscle strains.

Currently, some works deal with only the qualitative part of the strength training or weight lifting exercises [82, 68]. But Pernek *et al.* in [69], evaluates their system based on different weight loads for a set of upper body exercises. Though they deal with the quantitative observation of the strength training, it is not the objective

one rather their system incorporates a subjective measure of the intensity by asking the participants to self-assess the intensity of the exercise using the Borg's rating of perceived exertion (RPE). The motivation behind our idea are the limitations of existing works which are:

- Relies on multiple sensors [69, 68].
- Deals with only the qualitative part of the strength training [82, 68].
- Works with the subjective measurement of the intensity during quantitative analysis [69].

Therefore, in our work, we rely on a single sensor and focus on the quantitative part of the weight lifting activity. Our system is named PaWLA which is elaborated as PPG-based Weight Lifting Assessment.

5.2 Problem Definition

The strain sensing problem is physically a weight recognition problem where different weights have different strain effect on the users' muscles or body. Typically, a user lifts a certain amount of weight using his/her hand in a scenario for example, during workout or during any chores. Thus, the hand muscles of the user is tightened a bit which causes a certain amount of strain which might be tolerable or not tolerable to the user depending on the amount of weight of the object. Generally, a person knows the amount of weight he/she can comfortably lift and which amount of weight he/she can lift but has to suffer from strain effect or muscle fatigue.

The system should be such that it would be able to recognize the weight of the lifted object. Each user will have a certain predefined threshold for avoiding strenuous effect while lifting a weight. If the recognized weight is crossing the predefined threshold then the system should alert the user that the lifted object is going to put a strenuous effect on the user's muscles. Conventionally, a smart sensing system would

be applicable for addressing a problem like this where automation is required to perform an activity after recognizing the weights. The workflow for this type of system would as follows. The sensor will capture the user’s data which reflects the activity of lifting the object. Based on that captured data from the sensor, the system will be able to recognize the weight label. In order to achieve a good accuracy for a system as this, training data is required so that the model of the system can be trained well to classify the weight labels correctly.

The data for the system can be collected in a lot of ways such as, using additional monitoring hardware setup to get the sensing data (e.g., cameras, acoustic sonars, or custom-built sensors), putting smart sensors on the hand/arm, or leveraging the commercially off-the-shelf (COTS) smart devices (e.g., smartwatches or fitness-wristbands). In our work, the data is the sensor reading of Photoplethysmography sensor and is collected from the wrist-worn wearable device which contains the PPG sensor. The lifting of the object affecting the blood flow of the users’ wrist is taken into consideration. The strain effect would cause changes in the blood volume reflected upon the PPG reading samples over the time. Fig. 5.1 shows an example of two PPG readings for lifting two different weights.

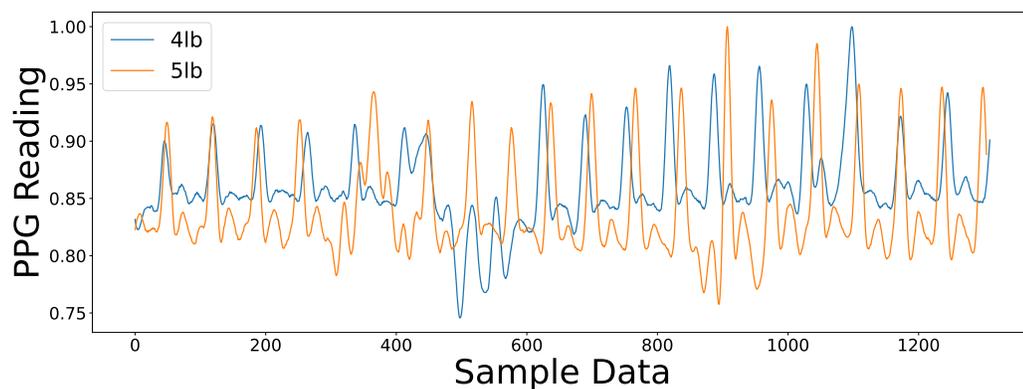


Figure 5.1: PPG sensor readings for lifting weights of 4 lb and 5 lb.

For the system to kick start, volunteers provide PPG sensing data of lifting objects of different weights while wearing the given wrist-worn wearable device. The system

is trained on that volunteered PPG data to create a model based on the features extracted from those weightlifting signals of varied weights. Then when a new signal input is given into the device, the system model based on the saved feature matrix decides which weight label the input signal falls into. It is mainly considered as a multi-class machine learning classification problem. A general picture of the process can be seen in Fig. 5.2. In case of PPG-based system, the input to the system model would be the PPG sensor's data.

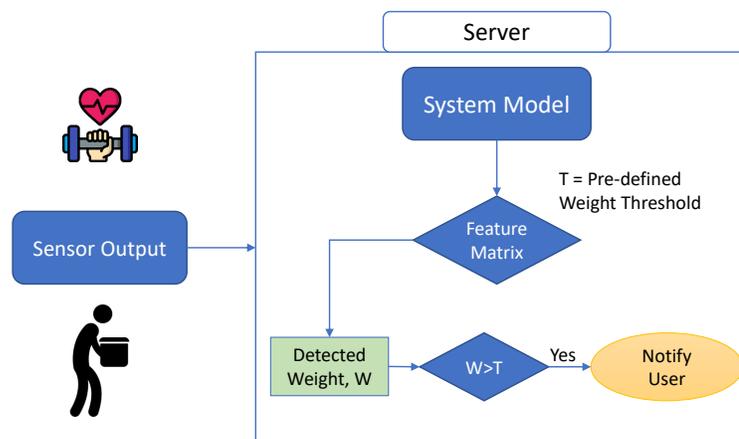


Figure 5.2: A generalized strain sensing problem.

5.3 System Design

In this section, we present the system design of our proposed work which includes the architecture and the workflow of the overall PaWLA system.

5.3.1 System Architecture

The system architecture of PaWLA is shown in Fig. 5.3, which consists of four parts: *Data Collection*, *Data Processing and Trimming*, *Feature Characterization*, and *Classification*. In the data collection module, when the user is lifting an object with a certain weight, the wrist-worn wearable device with a single PPG sensor is used to collect the PPG data. After the collection of the data, it is passed to the data

processing and trimming module where there are three steps of processing: noise filter, signal normalization, and signal trimming. Noise filter has the ability to filter out the combined noises of the signal. The signal normalization component normalizes any signal that comes into it. The normalized signals are trimmed in such a way that all the disturbances caused by the slide switch (used for starting/stopping data collection) are removed. After the data processing and trimming module, comes the feature characterization module which includes three main components: feature extraction, feature selection, and feature matrix. This module mainly deals with the extraction of relevant intrinsic characteristics of the input signal data that can discriminate each weight label from another. Finally, the last module is the classification module which is the foundation model of the system. The classification model is trained on the whole dataset of different weight labels in order to classify a new incoming signal into its corresponding proper weight labels.

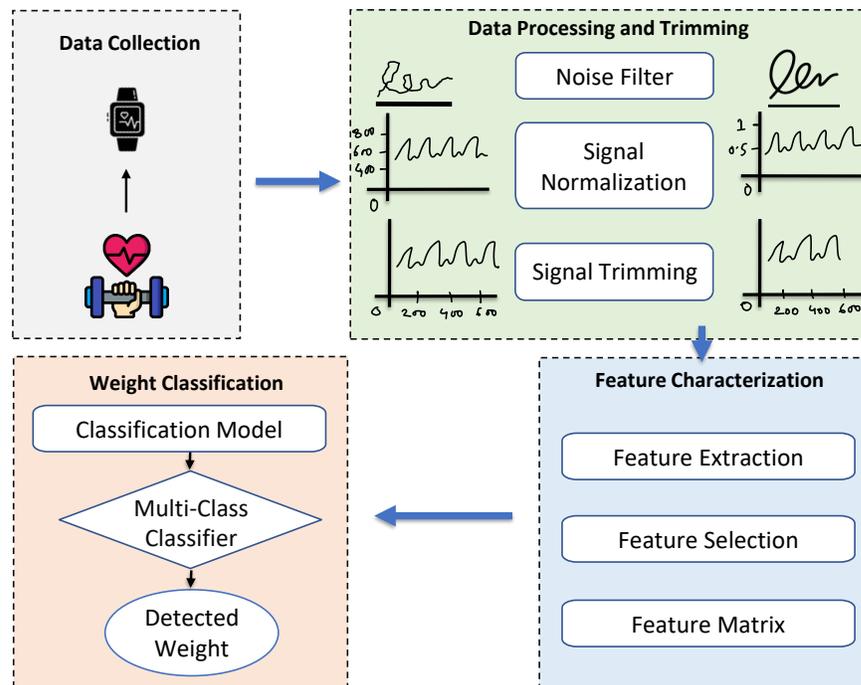


Figure 5.3: System architecture of PaWLA.

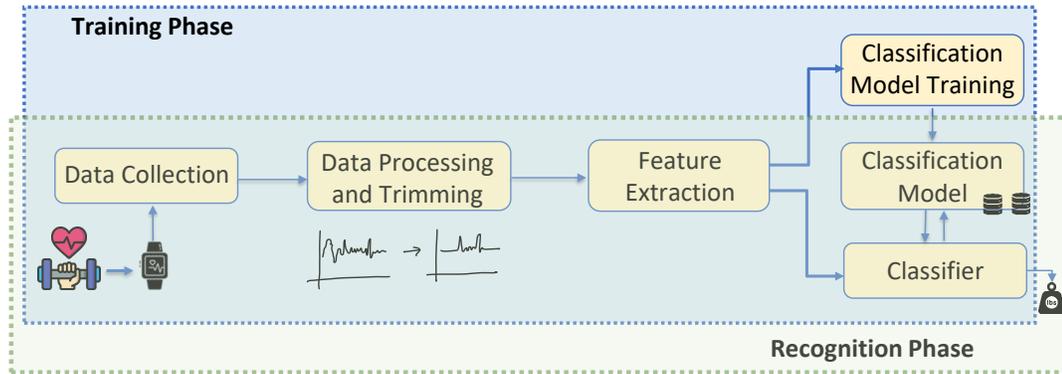


Figure 5.4: Workflow of PaWLA.

5.3.2 Workflow

The workflow of PaWLA is illustrated in Fig. 5.4. There are primarily two main phases for the system which are: *Training Phase*, and *Recognition Phase*. The workflow of the system in light of the phases is discussed below.

Training Phase: In the training phase, volunteers provide some sample weightlifting signals using the wrist-worn wearable device. The provided PPG sensor signal data is then sent to the server-end from the user-end. After the data collection, the data goes to the data processing and trimming module. The noise from the raw data is first filtered and normalized and passed on to the next step for trimming. After receiving the processed weightlifting data, features are extracted and filtered according to relevance in the feature characterization module mentioned in previous Section 5.3.1. Feature matrix is created based on the relevant features and stored in the database. The classifier is trained based on the feature matrix to create a model for each user. All the models are stored in the server for the recognition phase.

Recognition Phase: In the recognition phase, the data collection is done in the same way as in the training phase. The user trying to lift an object, would provide the PPG data while wearing the wrist-worn wearable device. The data would then be sent to the server-end for data processing which includes the noise filtering of the

raw data, normalizing the data, and trimming the signal data. After that, the feature matrix is generated based on the same feature extraction process mentioned in the training phase. Now, this feature matrix is passed on to the pre-trained classifier model for classifying the input data to its corresponding weight label. Then finally, the system lets the user know about the corresponding weight label for the lifted weight based on the comparison of the feature matrix of the newly collected data with the stored feature matrices in the database.

5.4 Challenges

Primarily, there are three challenges that needs to be addressed in order to recognize the lifted weights using the captured PPG signals from a wrist-worn device successfully. These are discussed one by one below.

5.4.1 Coarse-grained Wrist PPG Signals

This challenge is similar to the first challenge discussed in our first work in Section 3.4. Compared to electrocardiogram (ECG) signals, PPG signals suffer more noises and interference with other signals [1]. It is even more coarse-grained when the location of the PPG sensor is in the wrist region. In clinical settings, the conventional technique is to use the fingertip region-based PPG technique, as the critical landmarks are more detectable. It is relatively harder to detect the critical landmarks in the wrist region. This indicates the fact that the techniques used for the fingertip PPG systems will not be applicable to the wrist region PPG techniques. The stray effects of the AC current fields may cause power line interference due to the cable loops or issues in the electrodes. In Section 5.5, we will design our noise filtering module to tackle this challenge and extract critical landmarks. We also propose a feature extraction module in Section 5.6 to extract discriminating features to make the recognition system a fine-grained one.

5.4.2 Effect of PPG Sensor Placement

It is to be noted that the performance of a system can vary based on the placement of the PPG sensor as PPG readings vary at different locations. In order to observe the effects of placement of PPG sensor at different locations, we have built our own prototype where we can change the position of the PPG sensor to our desired locations. The sensor was placed around the Median Antebrachial Vein (bottom red circle in the right sub-figure of Fig. 3.7) in PPGSign [125], as the purpose of the system was to capture the minute movements of fingers along with the hand itself while writing the signature. Posterior wrist-side is one of the other locations for the placement which is the most common one for the existing smartwatches/fitness trackers. This location is the leftmost red circle in Fig. 3.7. As the lifting movement of the user is the same for all the weights, in our work, we do not focus on the signal caused by the weight lifting rather for our experiment, radial artery (the top red circle in the right sub-figure of Fig. 3.7) is the preferred location for the placement of the PPG sensor. The solution is feasible even though the location is not the commonly used one in a sense that the user can rotate their wrist-worn wearable device to the desired location before lifting the weights. In our experiment, we assume that the user is wearing the wrist-worn wearable device on the hand which is being used for the weight lifting activity.

5.4.3 Similarity in Readings for Nearby Weights

This is an intuitive challenge. In our work, we are focusing on the strain effect caused by the lifting of weights. In a practical scenario, the weights that are nearby each other will cause almost a similar strain effect on the user. As shown in Fig. 5.1, the PPG readings of the user lifting weights of 4 lb and 5 lb have a very close pattern. It is applicable for any nearby weights having similarity in readings. As we are analyzing the patterns of the PPG readings, it is a challenging task to classify them into correct labels. To do the classification accurately, traditional methods or techniques

are not sufficient enough. In our system, we have incorporated a feature characterization module which extracts discriminatory features from the input readings and classifies them into corresponding proper weight labels to solve this challenge even if the weights are close to each other.

5.5 Data Processing

In this section, we discuss the technical details for the data processing and trimming module mentioned earlier in the Section 5.3.1 for PaWLA. The module consists of *noise filtering*, *signal normalization*, and *signal trimming*.

5.5.1 Noise Filtering

There will always be noises in the raw PPG data collected via wrist-worn wearable devices because there are variations in surrounding environment and also changes in user's behavioral traits. Other than these common noises there are noises in the PPG sensor readings due to high frequency interference, power line interference, and baseline drifts because of the hardware imperfections. A straight forward solution would be applying a high band-pass filter or Butterworth filter with a high cut-off frequency at 2 Hz. But an issue is the filtration of probable important elements with a low frequency hidden within the input signal due to the lifting of the weight. In order to avoid that, Savitzky-Golay (S-G) filter [113] is applied to smooth the signal. S-G filter is a type of low-pass filter, which is a smoothing method based on local least-squares polynomial approximation. The polynomial fitting of the sample points and the evaluation of it at a single point within the approximation is equivalent to discrete convolution with a fixed impulse response [114]. S-G filter can not only remove the noises from the signal but also maintain the shape and height of the peaks of the signal's/sample's waveform. Fig. 3.10 shows an example of the effect of the S-G filter in our system. Before applying the filter, the signal contains granular noises along the trajectory (as shown in the left sub-figure), which is smoothed in the right sub-figure

after applying the S-G filter.

5.5.2 Signal Normalization

Different range of the PPG readings exist for different users. These PPG sensor readings generally ranges between 400 and 750. It is also possible that the same user starts showing different range of the PPG readings based on different physical conditions of the user. But if we focus on the visual pattern of the signals, usually they are the same for the same user. We want to capture the visual pattern of the signal rather than different range of values. For this reason, we normalize the signal with a range of 0 to 1 before passing the PPG data to the feature characterization module. If we focus on the pattern of the signals, the features extracted from those signals would be more consistent and help the system's classification accuracy eventually.

5.5.3 Signal Trimming

In our experimental setup, we have slide switch used for starting and stopping the data collection process. As a result of hardware imperfections generally, there are disturbances received in the sensor data due to the slide switch being turned on and off. As these disturbances are caused by the slide switch and does not reflect the real scenario of the user lifting a weight or putting a weight down, we need to get rid of these. Therefore, in order to get ride of these unnecessary disturbances, we performed a few micro seconds of signal trimming at the beginning and end of the input signal.

5.6 Feature Characterization

The processed signal is passed on to the feature characterization module from the data processing and trimming module. This module consists of *feature extraction*, *feature selection* and *feature matrix*. To facilitate the weight classification accurately, various PPG features are explored. The selected features are built into a feature matrix which helps to train the classifier required to classify the weights into the correct label. The flow of this module is given in Fig. 3.15.

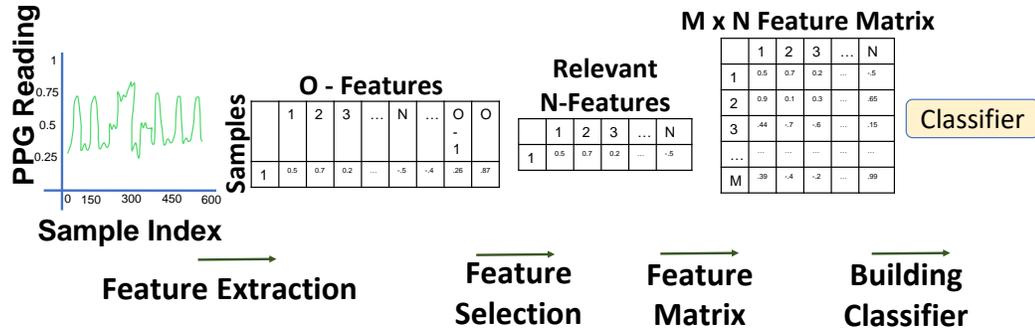


Figure 5.5: Flow in feature characterization module.

5.6.1 Feature Extraction

Time-domain features are first extracted from the clean PPG signal data. We use a time-series feature extraction tool, *tsfresh* [118], to do so. The tool is a python package to automatically calculate time series characteristics or features. It has 63 characterization methods to serve its purpose. For these methods, it is able to compute more than 700 time-series features in an accelerated way compared to existing time-consuming processes. The clean processed PPG signal data needs to be converted into suitable data frames so that it can be processed by *tsfresh*. For PaWLA, *tsfresh* is used to extract 787 time-domain features, such as absolute energy of the time series, sample entropy, number of different peaks, sample skewness, descriptive statistics on the auto-correlation of the time series, binned entropy of the power spectral density of the time series, and many more.

5.6.2 Feature Selection

In this step, we need to refine or filter our extracted features from the previous step and come up with only the most important features that can help discriminate the different weight labels. To obtain a proper selection of strong features is a hard problem for a time-series classification. A feature selection process based on scalable hypothesis tests is incorporated in *tsfresh*. Important features to be used for the machine learning model can be filtered beforehand by it. In the following lines we

tell how the process works. The feature vectors generated in the previous step are all individually tested to predict the labels in regards of their significance. The output of these tests gives a score vector for each feature vector. These score vectors are then evaluated on the basis of the Benjamini-Yekutieli procedure to decide which features to be discarded and which features are important enough to be kept [120]. Because of this process, around 400 (denoted by N) important features are selected as the relevant features out of the extracted 787 (denoted by O) features for PaWLA. The number of selected features N varies depending on different classification tasks (i.e. different weight categories mentioned in Table 6.1).

5.6.3 Feature Matrix

We then generate a feature matrix to represent all features of sampled signals from the user u . After collecting M calibrating sample signals s_i ($i = 1, \dots, M$) from this user, N relevant features $f_{i,j}$ ($j = 1, \dots, N$) are generated by *tsfresh* for each sample s_i . For each feature, its values $f_{i,j}$ might have a different range. Different machine learning algorithms are insensitive to the scale of diversified features. Therefore, we perform feature scaling on these values before passing them to train/feed the classifier. In our system, we use the following to standardize the feature value of $f_{i,j}$:

$$f'_{i,j} = \frac{f_{i,j} - \mu_j}{\sigma_j},$$

where $f_{i,j}$ is the original j -th feature's value of this sample s_i , and the new feature value $f'_{i,j}$ is calculated using the mean and standard deviation of all samples (i.e., $\mu_j = \frac{\sum_{i=1}^M f_{i,j}}{M}$ and $\sigma_j = \sqrt{\frac{\sum_{i=1}^M (f_{i,j} - \mu_j)^2}{M}}$). This feature scaling can lead to fast convergence for the machine learning algorithms. After standardizing the features, we form a $2D$ matrix of the dimension $M \times N$ for each user where each row represents the sample data and each column represents a selected relevant feature. This feature matrix is then used for training of the classification model of user u .

5.6.4 Classification

Similar to POSA, in PaWLA, we used several standard classifiers to perform the weight classification. The input of the classifier is the generated feature matrix, and the output is the corresponding weight label. In our experiments, we have tested on the following classifiers:

- Random Forest (RF)
- Support Vector Machine (RBF Kernel)
- Gradient Boosting (GB)
- k-nearest neighbor (kNN)

CHAPTER 6: PAWLA IMPLEMENTATION AND EVALUATION

In this chapter, we discuss the implementation of the so far discussed modules of the PaWLA system and the subsequent evaluation of the system’s different pre-mentioned classifiers based on the experiments.

6.1 Experimental Setup

The PaWLA prototype is quite similar to our POSA prototype except some extended accessories for the weight lifting scenario. It consists of *PaWLA Band*, *Weight Accessories*, and *Server*.

6.1.1 PaWLA Band

Though there are a number of commercially available fitness trackers and smart-watches in the market which leverages PPG sensors to measure the heartbeat/pulse of the user, in the present, most of the devices do not provide access to the raw data. We have developed our own prototype from off-the-shelf components to get hold of the raw data. We call our low-cost proof-of-concept smart band the PaWLA Band as shown in Fig. 6.1(a). It uses the same components as of the POSA band mentioned in Section 4.1.

The PaWLA band also consists of multiple components: an Arduino UNO micro-controller, a PPG sensor from World Famous Electronics, a velcro wristband, a USB cable to connect the micro-controller to the server, and a slide switch to start and stop the data collection process. The most commonly used LED in the COTS smart-watches and fitness trackers is the green LED. As green LED performs the best, we used the green LED in our prototype sensor. The PPG sensor is strapped to the Velcro band so that it remains attached to the wrist of the user when worn in order



Figure 6.1: PaWLA prototype and data collection: (a) hardware components of PaWLA Band; (b) weight accessories; (c) data collection with a PaWLA band.

to imitate the wrist-worn wearable devices.

6.1.2 Weight Accessories

Two accessories were considered for the weight lifting part of PaWLA as shown in Fig. 6.1(b). The first accessory is a small light bag which will contain the weights. And the second accessory is the weight to be considered. In our experimental setup, we considered dumbbells of different weights. There are two *2 lbs*, two *3 lbs*, and two *5 lbs* dumbbells.

6.1.3 Server

For the server side, similar to the one in Section 4.1.2, we use a Dell Inspiron 15 Laptop as the server. The real time data was transferred to the laptop (server) using a USB cable connected with the Arduino UNO micro-controller on one end and the other end with the laptop. To implement all the software module of PaWLA, Python 3.8 was selected. We utilized *tsfresh* tool [118] for the feature extraction task and the python package *scikit-learn* [121] for implementing the classifiers.

6.2 Data Collection

We have seen all the components of PaWLA in the previous sections. In this section, we make use of these components and discuss the process of collecting the PPG signal data from the users. As we have shown previously in Table 6.1, we are

Table 6.1: Distribution of Weights per Category.

Category	Weights
1 lb diff.	2 lb, 3 lb, 4 lb, 5 lb, 6 lb, 7 lb, 8 lb, 9 lb, 10 lb
2 lb diff.	2 lb, 4 lb, 6 lb, 8 lb, 10 lb or 3 lb, 5 lb, 7 lb, 9 lb
3 lb diff.	2 lb, 5 lb, 8 lb or 3 lb, 6 lb, 9 lb

considering weights ranging from 2 lb to 10 lb. We fill the bag, mentioned in the earlier section, with these variations of weights. The PaWLA band is wrapped around the wrist region with the PPG sensor facing towards the Radial Artery vein. At the beginning of the data collection process, the user stands in a stationary position as shown in Fig. 6.1(c). Then the slide switch is turned on to start the data collection process. The user stands still for a few seconds and then start lifting the bag for a few seconds and finally puts down the bag of weights afterwards. The whole time the real time data stream is transferred from the PPG sensor to the server using the Arduino UNO micro-controller via the USB cable. Then finally, the slide switched is turned off to stop the data collection process.

In our experiment, we recruited nine healthy participants (eight males and one female) from whom the PPG signals were collected. The participants were healthy in a sense that they had no history of heart disease and they were capable of lifting a bag with a maximum weight of 10 lbs. In our system, we considered nine weights starting from 2 lbs and ending at 10 lbs with 1 lb apart. For each weight label, the user provided 30 samples of data. So, in total there were 9 sessions of weight lifting for each user. We divided these sessions in to two different days for each user. The even weights were collected in the first day and the odd weights were collected in the second day. In total, we have collected $30 \times 9 \times 9 = 2,430$ samples of data.

6.3 Evaluation

We conduct weight classification tasks on different settings to extensively evaluate our PaWLA system. Primarily, three weight categories of training of weights are

considered for the evaluation, as shown in Table 6.1. All the evaluations for PaWLA are done on user-specific (user-dependent) models unless mentioned otherwise. User-specific (user-dependent) model training adoption means that each user has its own classifier model to classify the weight labels. Our primary focus for our evaluation is on the detection of the correct label of the weight lifted by the individual user. With different splitting size of training and testing data (ranging from 20% up to 80%), the data collected via *PaWLA Band* has been tested on multiple classifiers mentioned in Section 5.6.

6.3.1 Metrics

For the quantitative analysis of the system on certain weight classification tasks, we use similar metrics: *Precision*, *Recall*, and *F1 score*, as defined in Section 4.3.1. In addition, we also report

- *Confusion Matrix*: A square matrix, where each row and column represent the ground truth and the classification result, and each value in the matrix represents the percentage of a weight being classified into a weight label. The diagonal values is primarily focused during evaluation as it represents the correctly predicted labels.

6.3.2 Performance

The overall average F1 score of our system irrespective of the weight categories is around 91.8% with training size of 80% data. Fig. 6.2 shows the confusion matrix of the overall performance of the system being trained with weights with 2 *lb* difference i.e. the even and odd weights (2 *lb*, 4 *lb*, 6 *lb*, 8 *lb*, 10 *lb* and 3 *lb*, 5 *lb*, 7 *lb*, 9 *lb*). For the even weights, our system can recognize the weight correctly at an average of 85.9%. For the odd weights, it can do it at an average of 88.4%.

We also perform experiments to investigate the effects of different folds of our system, such as different classifiers, training sizes, sensor placements, weight categories,

and user independent classification.

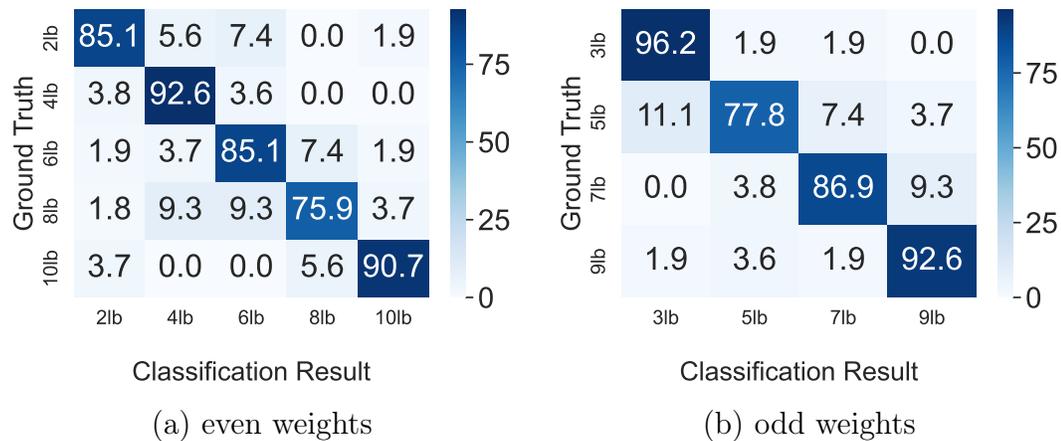


Figure 6.2: Overall performance of classification for (a) even and (b) odd weights.

Impact of Classifiers: Fig. 6.3(a) presents a comparison of the performances of four classifiers mentioned in Section 5.6. Here, we report the average performances over 9 volunteers with the weight category of 2 *lb* differences and 60% data as training data. Among the four classifiers, Random Forest (RF) and Support Vector Machine (SVM) perform the best. Therefore, for the remaining evaluations, we use Random Forest as the default classifier.

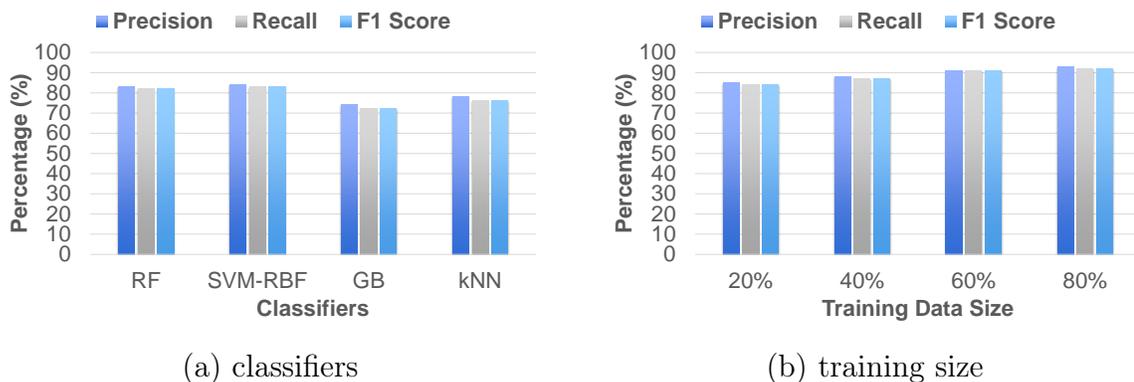


Figure 6.3: Performance of PaWLA with (a) different classifiers (weight category of 2 *lb* difference, 60% data as training data) (b) different training sizes varying within 20 – 80% of data (average among all weight categories).

Impact of Training Sizes: To study the impact of training data sizes, we train

our model with different partitions of the collected data (i.e, using 20% to 80% as the training data). The average performance for each training size is given in Fig. 6.3(b). Here, the results are average among all weight categories. We observe that with the increase in training size, the average F1 score also increases with a maximum of 91.8%. Even with the lowest training size (20% of data) the average F1 score of our system is still 84.2%.

Impact of Sensor Placement: As discussed in Section 5.4, PPG sensor placement also plays an important role. We test two sensor placements in our experiments: radial artery and posterior wrist side. For this experiment, we test our user-specific model with one user, a training size of 60%, and all the weight categories. As shown in Fig. 6.4(a), the average F1 score of PaWLA is 94.6% and 91.6% when the sensor location is at radial artery and posterior wrist side respectively. This shows a promise that our system can be implemented in COTS smartwatches/fitness watches where the PPG sensor is commonly placed over the posterior wrist side region.

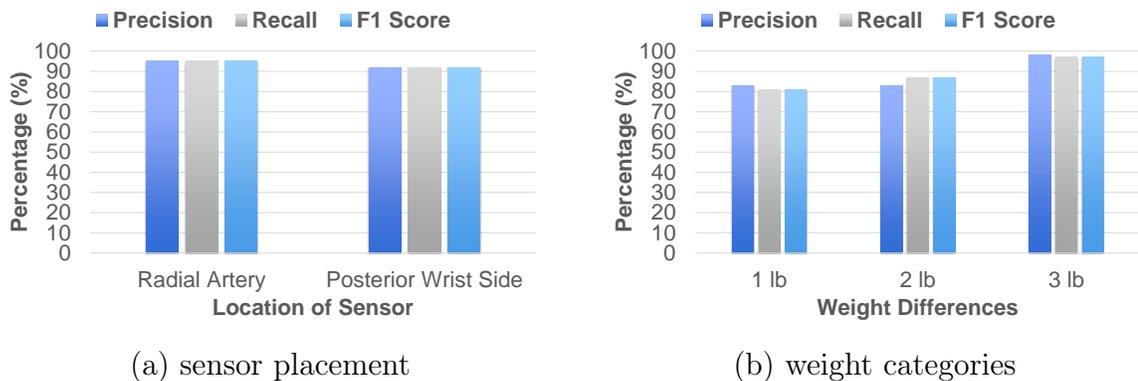


Figure 6.4: Performance of PaWLA for (a) different sensor placements (60% as training data) (b) different weight differences (80% as training data).

Impact of Weight Differences: We test our system with different weight categories mentioned in Table 6.1 with a training size of 80%. As shown in Fig. 6.4(b), PaWLA performs the best when the training is done with the weights being 3 *lb* apart with an average F1 score of 97.4%. The system performs the least with an average F1 score of 80.9% when the weights are only 1 *lb* apart because of the similar

strains caused by nearby weights. This shows that our current system can achieve good accuracy when the weight difference is at least 2 *lb* apart. We leave further investigation on new techniques to recognize finer weight difference.

User Dependent Model vs User Independent Model: So far, in our experiment, we focus on user-dependent model where a user-specific model is trained and tested for each of the participant. Fig. 6.5 shows the detailed performance of each user-specific model for 9 participants when the training data size is 80%. Overall, the model performance is similar irrespective of different users with an average of 91.8% F1 score, a maximum of 95.0% and a minimum of 88.0% F1 score. Then we also consider a user independent model where a global model is trained and tested using data from all users. In other word, the global model is trained for each weight label over an aggregation of all the users' data for that weight label. The performance of the user independent model is shown in Table 6.2. The average F1 score is now 80.0% with a maximum of 92.5% when the weight difference between the labels is 3 *lb*. This proves the feasibility of using our pre-trained system for users without any training data at a standard performance or at least as a starting point. With more user-specific training, the performance can then be further improved towards the user dependent model.

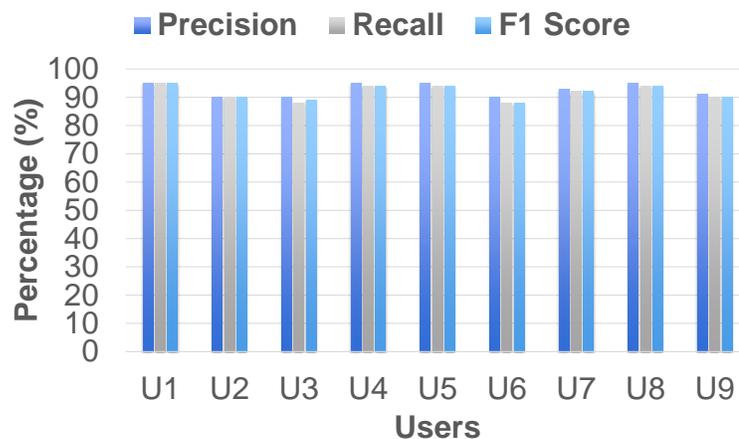


Figure 6.5: Performance of user specific model for each user (80% training).

Table 6.2: Performance of User Independent Model (80% training).

Weight Category	Precision	Recall	F1 Score
1 <i>lb</i> difference	74.0%	73.0%	74.0%
2 <i>lb</i> difference	73.5%	73.5%	73.5%
3 <i>lb</i> difference	92.5%	92.5%	92.5%
average over all	80.0%	79.7%	80.0%

CHAPTER 7: CONCLUSION AND FUTURE WORKS

7.1 Conclusion

In my dissertation, we firstly address the offline/online signature authentication problem using the Photoplethysmography sensor present in the wrist-worn wearable device. Almost all the smartwatches/fitness trackers have this PPG sensor as a built-in sensor. For which there have been research with PPG sensor for being a biometric authentication modality. Regarding signature authentication, most related works rely on dedicated devices/additional hardware. Also existing solutions do not protect against the common attack models. They are also not very user friendly or non-intrusive. The proposed system (POSA) is non-intrusive, secure, and low-cost which makes it a reliable solution. It can be used in banking sectors, agreement signing scenarios or in any smart device sign in. It could be used in situations where an authentication of signature, whether it is online or offline, is needed. Our system POSA leverages the PPG sensed data from the wrist-worn wearable which is mainly a measure of the volumetric variation in the blood flow when the user is actually giving his/her signature. The sensed data is then processed in its data processing module where the data is first noise filtered and normalized, then the signature signal portion is segmented and finally trained with different classifiers based on the feature matrix generated in the feature characterization module. The experiments show that the system is promising and feasible enough to be implemented commercially and thus solve the offline/online signature authentication problem.

In the second work, we worked on a strain sensing problem which is physically a weight recognition problem. There have been a significant amount of research in sensor-based fitness applications but quantitative assessment of physical activities is

yet to be explored. In this work, we focus on the quantitative side of assessing a physical activity. The proposed system (PaWLA) is non-intrusive, low-cost, and reliable based on the extensive evaluation results. PaWLA leverages the PPG sensed data from the wrist region of the user lifting a certain weight. The sensed data is then passed on to the data processing module where it is noise filtered, normalized and trimmed. After that the processed signal is moved on to the feature characterization module where discriminating features are extracted and then relevant features are selected to build the feature matrix. The classifiers are trained on these feature matrices to classify the lifted weights to corresponding weight labels. As it exploits the blood flow change within the wrist region based on the PPG sensor, it can be implemented with the existing smartwatches/fitness trackers. Evaluation of the system with nine volunteers shows that PaWLA can achieve a very good performance proving the feasibility and efficiency of the proposed method.

7.2 Future Works

With the advancement of Internet of Things (IoT), smart technologies are being innovated, leading to the success of smart sensing field. Automation in different domains is taking place exponentially. Different applications or systems are designed for different set of tasks. In this dissertation, we introduced POSA and PaWLA one of which focused on a signature authentication-based problem and another on a strain-sensing problem respectively. There are rooms for improvement for both the systems. For our future work beyond the dissertation, we plan to introduce two systems - Multi-modal Offline/Online Signature Authentication (MOSA) and PPG-based Application for Strain Sensing (PASS).

Multi-modal Offline/Online Signature Authentication (MOSA): So far, throughout Chapters 3 and 4, the PPG sensor-based offline/online signature authentication system was discussed. Though the system is low-cost, non-intrusive, and secure, it is not fully robust enough. To further extend this work and develop a more

robust system, we have come up with Multi-modal Offline/Online Signature Authentication (MOSA). In this work, we will make use of the accelerometer as the second modality for authentication. For this, we would need such a wrist-worn wearable device that has both the PPG sensor and the tri-axial accelerometer sensor. Using two sources would give us the better chance to separate the mixed signals and also verify the user's signature more robustly. Another modality that we want to pursue is the magnetometer within a wearable device. The signals from it would interfere with the magnetic signals of a smartphone leading it to change in the received signal which can help detect the input by the user to authenticate the signatures being written. If we combine all the modalities for the signature authentication, the system will be able to handle situations when one of the modalities fail. Thus, we can say that the multi-modal offline/online signature authentication is more robust.

PPG-based Application for Strain Sensing (PASS): Another future work that we are planning to work on is the PPG-based Application for Strain Sensing (PASS) which is an extension to PaWLA. In PaWLA, we tested out the global model which was performing well but the individual model was performing better. For PASS, we firstly plan to collect more data from diverse participants with different genders, ages, and skin colors to train the global model on a more extensive dataset. We also want to make the system robust enough to handle different states of a participant such as, emotions, pre-workout, post-workout, cardiac disease. In PASS, we want to investigate a neural network-based implementation to make the global model perform better for fine weight differences. Generating a universal classifier to be able to sense the strain for all the users depending on the predefined weight threshold of the user is another goal for PASS. Lastly, for our future work, we will make efforts to explore a real implementation of the PASS system on commercially off-the-shelf smartwatches/fitness trackers with a user-friendly mobile and wearable app.

REFERENCES

- [1] T. Zhao, Y. Wang, J. Liu, Y. Chen, J. Cheng, and J. Yu, “Trueheart - continuous authentication on wrist-worn wearables using ppg-based biometrics,” in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 2020.
- [2] B. Spencer, M. Ruiz Sandoval, and N. Kurata, “Smart sensing technology: Opportunities and challenges,” *Structural Control and Health Monitoring*, vol. 11, pp. 349 – 368, 10 2004.
- [3] N. Clarke and S. Furnell, “Advanced user authentication for mobile devices,” *Computers & Security*, vol. 26, no. 2, pp. 109 – 119, 2007.
- [4] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, “A survey on security and privacy issues in internet-of-things,” *IEEE Internet of Things Journal*, vol. 4, pp. 1250–1258, Oct 2017.
- [5] E. G.-H. C Cadwalladr, “Revealed: 50 million facebook profiles harvested for cambridge analytica in major data breach,” *The Guardian*, 2018.
- [6] J. Wang, T. Herath, R. Chen, A. Vishwanath, and H. R. Rao, “Research article phishing susceptibility: An investigation into the processing of a targeted spear phishing email,” *IEEE Transactions on Professional Communication*, vol. 55, no. 4, pp. 345–362, 2012.
- [7] J. Erickson, *Hacking: The Art of Exploitation, 2nd Edition*. USA: No Starch Press, second ed., 2008.
- [8] S.-W. Lee, D.-K. Woo, Y.-K. Son, and P.-S. Mah, “Wearable bio-signal(ppg)-based personal authentication method using random forest and period setting considering the feature of ppg signals,” *Journal of Computers*, vol. 14(4), pp. 283–294, 2019.
- [9] C. Wu, K. He, J. Chen, Z. Zhao, and R. Du, “Liveness is not enough: Enhancing fingerprint authentication with behavioral biometrics to defeat puppet attacks,” in *29th USENIX Security Symposium (USENIX Security 20)*, (Boston, MA), USENIX Association, Aug. 2020.
- [10] S. Ahmed, A. R. Chowdhury, K. Fawaz, and P. Ramanathan, “Preech: A system for privacy-preserving speech transcription,” in *29th USENIX Security Symposium (USENIX Security 20)*, USENIX Association, Aug. 2020.
- [11] Q. Wang, X. Lin, M. Zhou, Y. Chen, C. Wang, Q. Li, and X. Luo, “Voicepop: A pop noise based anti-spoofing system for voice authentication on smartphones,” in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 2062–2070, 2019.

- [12] J. Liu, C. Shi, Y. Chen, H. Liu, and M. Gruteser, “Cardiocam: Leveraging camera on mobile devices to verify users while their heart is pumping,” in *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '19, (New York, NY, USA), pp. 249–261, Association for Computing Machinery, 2019.
- [13] S. Li, A. Ashok, Y. Zhang, C. Xu, J. Lindqvist, and M. Gruteser, “Whose move is it anyway? authenticating smart wearable devices using unique head movement patterns,” in *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 1–9, 2016.
- [14] J. Liu, Y. Chen, Y. Dong, Y. Wang, T. Zhao, and Y. Yao, “Continuous user verification via respiratory biometrics,” in *2020 IEEE International Conference on Computer Communications (INFOCOM)*, 2020.
- [15] S. Lee, K. N. Ha, and K. C. Lee, “A pyroelectric infrared sensor-based indoor location-aware system for the smart home,” *IEEE Transactions on Consumer Electronics*, vol. 52, no. 4, pp. 1311–1317, 2006.
- [16] B. George, H. Zangl, T. Bretterkieber, and G. Brasseur, “A combined inductive-capacitive proximity sensor for seat occupancy detection,” *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 5, pp. 1463–1470, 2010.
- [17] Jeongho Kang, Yongduk Kim, Hyungpyo Kim, Junwhan Jeong, and Sekwang Park, “Comfort sensing system for indoor environment,” in *Proceedings of International Solid State Sensors and Actuators Conference (Transducers '97)*, vol. 1, pp. 311–314 vol.1, 1997.
- [18] Y. Shu, C. Li, Z. Wang, W. Mi, Y. Li, and T.-L. Ren, “A pressure sensing system for heart rate monitoring with polymer-based pressure sensors and an anti-interference post processing circuit,” *Sensors*, vol. 15, pp. 3224–3235, Feb 2015.
- [19] A. Pandharipande and D. Caicedo, “Smart indoor lighting systems with luminaire-based sensing: A review of lighting control approaches,” *Energy and Buildings*, vol. 104, pp. 369 – 377, 2015.
- [20] J. M. Sim, Y. Lee, and O. Kwon, “Acoustic sensor based recognition of human activity in everyday life for smart home services,” *International Journal of Distributed Sensor Networks*, vol. 11, no. 9, p. 679123, 2015.
- [21] M. Popa, “Hand gesture recognition based on accelerometer sensors,” in *The 7th International Conference on Networked Computing and Advanced Information Management*, pp. 115–120, 2011.
- [22] A. Pandharipande and S. Li, “Light-harvesting wireless sensors for indoor lighting control,” *IEEE Sensors Journal*, vol. 13, no. 12, pp. 4599–4606, 2013.

- [23] G. Krishnamurthy and M. Ghovanloo, "Tongue drive: a tongue operated magnetic sensor based wireless assistive technology for people with severe disabilities," in *2006 IEEE International Symposium on Circuits and Systems*, pp. 4 pp.–, 2006.
- [24] J. Liu, C. Wang, Y. Chen, and N. Saxena, "Vibwrite: Towards finger-input authentication on ubiquitous surfaces via physical vibration," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS '17*, (New York, NY, USA), pp. 73–87, Association for Computing Machinery, 2017.
- [25] O. Postolache, M. Pereira, and P. Girao, "Smart sensor network for air quality monitoring applications," in *2005 IEEE Instrumentation and Measurement Technology Conference Proceedings*, vol. 1, pp. 537–542, 2005.
- [26] A. B. M. M. Rahman, T. Li, and Y. Wang, "Recent advances in indoor localization via visible lights: A survey," *Sensors*, vol. 20, p. 1382, Mar 2020.
- [27] H. Askari, A. Khajepour, M. B. Khamesee, and Z. L. Wang, "Embedded self-powered sensing systems for smart vehicles and intelligent transportation," *Nano Energy*, vol. 66, p. 104103, 2019.
- [28] M. Rath, "Smart traffic management system for traffic control using automated mechanical and electronic devices," *IOP Conference Series: Materials Science and Engineering*, vol. 377, p. 012201, jun 2018.
- [29] S. Majumder and M. J. Deen, "Smartphone sensors for health monitoring and diagnosis," *Sensors*, vol. 19, p. 2164, May 2019.
- [30] M. E. E. Alahi, L. Xie, S. Mukhopadhyay, and L. Burkitt, "A temperature compensated smart nitrate-sensor for agricultural industry," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 9, pp. 7333–7341, 2017.
- [31] S. K. Bolisetti, M. Patwary, A. Soliman, and M. Abdel-Maguid, "Rf sensing based target detector for smart sensing within internet of things in harsh sensing environments," *IEEE Access*, vol. 5, pp. 13346–13363, 2017.
- [32] A. Shahzad, M. Lee, N. Xiong, G. Jeong, Y.-K. Lee, J.-Y. Choi, A. Mahesar, and I. Ahmad, "A secure, intelligent, and smart-sensing approach for industrial system automation and transmission over unsecured wireless networks," *Sensors*, vol. 16, p. 322, Mar 2016.
- [33] M. Srivastava, R. Muntz, and M. Potkonjak, "Smart kindergarten: Sensor-based wireless networks for smart developmental problem-solving environments," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, MobiCom '01*, (New York, NY, USA), pp. 132–138, Association for Computing Machinery, 2001.

- [34] S. Pirbhulal, H. Zhang, M. E Alahi, H. Ghayvat, S. Mukhopadhyay, Y.-T. Zhang, and W. Wu, "A novel secure iot-based smart home automation system using a wireless sensor network," *Sensors*, vol. 17, p. 69, Dec 2016.
- [35] Y. Zeng, P. H. Pathak, and P. Mohapatra, "Wiwho: Wifi-based person identification in smart spaces," in *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 1–12, 2016.
- [36] W. Yang, S. Wang, J. Hu, Z. Guanglou, and C. Valli, "Security and accuracy of fingerprint-based biometrics: A review," *Symmetry*, vol. 11, p. 141, 01 2019.
- [37] H. Wang, D. Lymberopoulos, and J. Liu, "Sensor-based user authentication," in *2015 Wireless Sensor Networks: 12th European Conference (EWSN)*, vol. 8965, pp. 168–185, 02 2015.
- [38] L. Lu, J. Yu, Y. Chen, H. Liu, Y. Zhu, L. Kong, and M. Li, "Lip reading-based user authentication through acoustic sensing on smartphones," *IEEE/ACM Transactions on Networking*, vol. 27, no. 1, pp. 447–460, 2019.
- [39] M. Ehatisham-ul Haq, M. A. Azam, J. Loo, K. Shuang, S. Islam, U. Naeem, and Y. Amin, "Authentication of smartphone users based on activity recognition and mobile sensing," *Sensors*, vol. 17, p. 2043, Sep 2017.
- [40] D. Wang, Y. Zhang, C. Yao, J. Wu, H. Jiao, and M. Liu, "Toward force-based signature verification: A pen-type sensor and preliminary validation," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 4, pp. 752–762, 2010.
- [41] J. BROMLEY, J. W. BENTZ, L. BOTTOU, I. GUYON, Y. LECUN, C. MOORE, E. SÄCKINGER, and R. SHAH, "Signature verification using a "siamese" time delay neural network," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 07, no. 04, pp. 669–688, 1993.
- [42] A. Shastry, R. Burchfield, and S. Venkatesan, "Dynamic signature verification using embedded sensors," in *2011 International Conference on Body Sensor Networks*, pp. 168–173, 2011.
- [43] D. Arjun, P. K. Indukala, and K. A. U. Menon, "Border surveillance and intruder detection using wireless sensor networks: A brief survey," in *2017 International Conference on Communication and Signal Processing (ICCSP)*, pp. 1125–1130, 2017.
- [44] M. A. A. Al-qaness, F. Li, X. Ma, and G. Liu, "Device-free home intruder detection and alarm system using wi-fi channel state information," *International Journal of Future Computer and Communication*, vol. 5, pp. 180–186, 01 2016.
- [45] F. Alsalami, Z. Ahmad, S. Zvanovec, P. Haigh, O. Haas, and S. Rajbhandari, "Indoor intruder tracking using visible light communications," *Sensors*, vol. 19, p. 4578, Oct 2019.

- [46] D. Castaneda, A. Esparza, M. Ghamari, C. Soltanpur, and H. Nazeran, "A review on wearable photoplethysmography sensors and their potential future applications in health care," *International journal of biosensors & bioelectronics*, vol. 4, no. 4, pp. 195–202, 2018. PMC6426305[pmcid].
- [47] T. Tamura, Y. Maeda, M. Sekine, and M. Yoshida, "Wearable photoplethysmographic sensors-past and present," *Electronics*, vol. 3, pp. 282–302, Apr 2014.
- [48] G. Fortino and V. Giampà, "Ppg-based methods for non invasive and continuous blood pressure measurement: an overview and development issues in body sensor networks," in *2010 IEEE International Workshop on Medical Measurements and Applications*, pp. 10–13, 2010.
- [49] G. Lu, F. Yang, J. A. Taylor, and J. F. Stein, "A comparison of photoplethysmography and ecg recording to analyse heart rate variability in healthy subjects," *Journal of Medical Engineering & Technology*, vol. 33, no. 8, pp. 634–641, 2009. PMID: 19848857.
- [50] S. Bagha and L. Shaw, "A real time analysis of ppg signal for measurement of spo2 and pulse rate," *INTERNATIONAL JOURNAL OF COMPUTER APPLICATIONS*, 12 2011.
- [51] M. Elgendi, R. Fletcher, Y. Liang, N. Howard, N. H. Lovell, D. Abbott, K. Lim, and R. Ward, "The use of photoplethysmography for assessing hypertension," *npj Digital Medicine*, vol. 2, p. 60, Jun 2019.
- [52] Y. Mendelson and C. Pujary, "Measurement site and photodetector size considerations in optimizing power consumption of a wearable reflectance pulse oximeter," in *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No.03CH37439)*, vol. 4, pp. 3016–3019 Vol.4, 2003.
- [53] M. Poh, N. C. Swenson, and R. W. Picard, "Motion-tolerant magnetic earring sensor and wireless earpiece for wearable photoplethysmography," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 3, pp. 786–794, 2010.
- [54] K. Budidha and P. A. Kyriacou, "The human ear canal: investigation of its suitability for monitoring photoplethysmographs and arterial oxygen saturation," *Physiol Meas*, vol. 35, pp. 111–128, Feb 2014.
- [55] Y. Lee, H. Shin, J. Jo, and Y. Lee, "Development of a wristwatch-type ppg array sensor module," in *2011 IEEE International Conference on Consumer Electronics -Berlin (ICCE-Berlin)*, pp. 168–171, 2011.
- [56] S. S. Thomas, V. Nathan, C. Zong, K. Soundarapandian, X. Shi, and R. Jafari, "Biowatch: A noninvasive wrist-based blood pressure monitor that incorporates training techniques for posture and subject variability," *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 5, pp. 1291–1300, 2016.

- [57] J. Spooren, D. Preuveneers, and W. Joosen, "Ppg2live: Using dual ppg for active authentication and liveness detection," in *2019 International Conference on Biometrics (ICB)*, pp. 1–6, 2019.
- [58] N. Karimian, M. Tehranipoor, and D. Forte, "Non-fiducial ppg-based authentication for healthcare application," in *2017 IEEE EMBS International Conference on Biomedical Health Informatics (BHI)*, pp. 429–432, 2017.
- [59] J. Sancho, A. Alesanco, and J. García, "Biometric authentication using the ppg: A long-term feasibility study," *Sensors (Basel, Switzerland)*, vol. 18, 05 2018.
- [60] A. Reşit Kavsaoglu, K. Polat, and M. Recep Bozkurt, "A novel feature ranking algorithm for biometric recognition with ppg signals," *Computers in Biology and Medicine*, vol. 49, pp. 1 – 14, 2014.
- [61] L. Everson, D. Biswas, M. Panwar, D. Rodopoulos, A. Acharyya, C. H. Kim, C. Van Hoof, M. Konijnenburg, and N. Van Helleputte, "Biometricnet: Deep learning based biometric identification using wrist-worn ppg," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2018.
- [62] T. Ohtsuki and H. Kamoi, "Biometric authentication using hand movement information from wrist-worn ppg sensors," in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1–5, 2016.
- [63] J. Shang and J. Wu, "A usable authentication system using wrist-worn photoplethysmography sensors on smartwatches," in *2019 IEEE Conference on Communications and Network Security (CNS)*, pp. 1–9, 2019.
- [64] Y. Cao, Q. Zhang, F. Li, S. Yang, and Y. Wang, "Ppgpass - nonintrusive and secure mobile two-factor authentication via wearables," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 2020.
- [65] T. Choudhary and M. S. Manikandan, "Robust photoplethysmographic (ppg) based biometric authentication for wireless body area networks and m-health applications," in *2016 Twenty Second National Conference on Communication (NCC)*, pp. 1–6, 2016.
- [66] N. Karimian, "Cardiovascular PPG biometric key generation for IoT in healthcare domain," in *Mobile Multimedia/Image Processing, Security, and Applications 2019* (S. S. Agaian, V. K. Asari, and S. P. DelMarco, eds.), vol. 10993, pp. 202 – 208, International Society for Optics and Photonics, SPIE, 2019.
- [67] J. Yathav, A. Bailur, A. K. Goyal, and Abhinav, "mibeat based continuous and robust biometric identification system for on-the-go applications," in *Proceedings of International Conference on Communication and Networks* (N. Modi, P. Verma, and B. Trivedi, eds.), (Singapore), pp. 269–275, Springer Singapore, 2017.

- [68] J. Qi, P. Yang, M. Hanneghan, S. Tang, and B. Zhou, “A hybrid hierarchical framework for gym physical activity recognition and measurement using wearable sensors,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1384–1393, 2019.
- [69] I. Pernek, G. Kurillo, G. Stiglic, and R. Bajcsy, “Recognizing the intensity of strength training exercises with wearable sensors,” *Journal of Biomedical Informatics*, vol. 58, pp. 145–155, 2015.
- [70] K.-H. Chang, M. Y. Chen, and J. Canny, “Tracking free-weight exercises,” in *Proceedings of the 9th International Conference on Ubiquitous Computing, UbiComp '07*, (Berlin, Heidelberg), p. 19–37, Springer-Verlag, 2007.
- [71] M. Luzar, A. Czajkowski, R. Kasperowicz, M. Rot, M. Semegen, and J. Korbicz, “Sensor-based weightlifting workout assisting system design and its practical implementation,” *Journal of Sensors*, vol. 2019, 2019.
- [72] S. Milanko and S. Jain, “LiftRight: Quantifying strength training performance using a wearable sensor,” *Smart Health*, vol. 16, p. 100115, 2020.
- [73] Y. Kowsar, E. Velloso, L. Kulik, and C. Leckie, “LiftSmart: a monitoring and warning wearable for weight trainers,” in *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, pp. 298–301, 2019.
- [74] C. Shen, B.-J. Ho, and M. Srivastava, “Milift: Efficient smartwatch-based workout tracking using automatic segmentation,” *IEEE Transactions on Mobile Computing*, vol. 17, no. 7, pp. 1609–1622, 2017.
- [75] Y. Xie, F. Li, Y. Wu, and Y. Wang, “HearFit+: Acoustic-based fitness monitoring using smart speakers,” *IEEE Transactions on Mobile Computing*, 2021.
- [76] Y. Xie, F. Li, Y. Wu, and Y. Wang, “HearFit: Fitness monitoring using acoustic sensing on smart speakers,” in *Proceedings of IEEE 40th Conference on Computer Communications (INFOCOM 2021)*, 2021.
- [77] N. Xiao, P. Yang, Y. Yan, H. Zhou, and X.-Y. Li, “Motion-Fi: Recognizing and counting repetitive motions with passive wireless backscattering,” in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pp. 2024–2032, IEEE, 2018.
- [78] K.-W. Ai, Z.-Y. Bi, and X. Han, “The development of a real-time feedback system in weightlifting,” in *ISBS-Conference Proceedings Archive*, 2014.
- [79] A. Yasser, D. Tariq, R. Samy, M. Allah, and A. Atia, “Smart coaching: Enhancing weightlifting and preventing injuries,” *International Journal of Advanced Computer Science and Applications*, vol. 10, p. 01, 2019.

- [80] P. Madanayake, W. Wickramasinghe, H. Liyanarachchi, H. Herath, A. Karunasena, and T. Perera, “Fitness Mate: Intelligent workout assistant using motion detection,” in *2016 IEEE International Conference on Information and Automation for Sustainability (ICIAfS)*, pp. 1–5, IEEE, 2016.
- [81] E. Velloso, A. Bulling, and H. Gellersen, “Motionma: Motion modelling and analysis by demonstration,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1309–1318, 2013.
- [82] E. Velloso, A. Bulling, H. Gellersen, W. Ugulino, and H. Fuks, “Qualitative activity recognition of weight lifting exercises,” in *Proceedings of the 4th Augmented Human International Conference, AH '13*, (New York, NY, USA), p. 116–123, Association for Computing Machinery, 2013.
- [83] J. Paay, J. Kjeldskov, F. Sorensen, T. Jensen, and O. Tirosh, “Weight-Mate: Adaptive training support for weight lifting,” in *Proceedings of the 31st Australian Conference on Human-Computer-Interaction*, pp. 95–105, 2019.
- [84] Y. Zou, D. Wang, S. Hong, R. Ruby, D. Zhang, and K. Wu, “A low-cost smart glove system for real-time fitness coaching,” *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7377–7391, 2020.
- [85] X. Guo, J. Liu, and Y. Chen, “FitCoach: Virtual fitness coach empowered by wearable mobile devices,” in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pp. 1–9, IEEE, 2017.
- [86] D. S. Elvitigala, D. J. Matthies, L. David, C. Weerasinghe, and S. Nanayakkara, “GymSoles: Improving squats and dead-lifts by visualizing the user’s center of pressure,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–12, 2019.
- [87] F. Xiao, J. Chen, X. Xie, L. Gui, L. Sun, and R. Wang, “SEARE: A system for exercise activity recognition and quality evaluation based on green sensing,” *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 3, pp. 752–761, 2020.
- [88] M. B. Yilmaz, B. Yanikoglu, C. Tirkaz, and A. Kholmatov, “Offline signature verification using classifier combination of hog and lbp features,” in *2011 International Joint Conference on Biometrics (IJCB)*, pp. 1–7, 2011.
- [89] B. Fang, C. Leung, Y. Tang, P. Kwok, K. Tse, and Y. Wong, “Offline signature verification with generated training samples,” *IEE Proceedings - Vision, Image and Signal Processing*, vol. 149, pp. 85–90(5), April 2002.
- [90] M. A. Ferrer, J. F. Vargas, A. Morales, and A. Ordonez, “Robustness of offline signature verification based on gray level features,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 966–977, 2012.

- [91] C. Santos, E. J. R. Justino, F. Bortolozzi, and R. Sabourin, "An off-line signature verification method based on the questioned document expert's approach and a neural network classifier," in *Ninth International Workshop on Frontiers in Handwriting Recognition*, pp. 498–502, 2004.
- [92] A. Alaei, S. Pal, U. Pal, and M. Blumenstein, "An efficient signature verification method based on an interval symbolic representation and a fuzzy similarity measure," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 10, pp. 2360–2372, 2017.
- [93] N. Sae-Bae and N. Memon, "Online signature verification on mobile devices," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 6, pp. 933–947, 2014.
- [94] B. L. Van, S. Garcia-Salicetti, and B. Dorizzi, "On using the viterbi path along with hmm likelihood information for online signature verification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 5, pp. 1237–1247, 2007.
- [95] E. Maiorana, P. Campisi, J. Fierrez, J. Ortega-Garcia, and A. Neri, "Cancelable templates for sequence-based biometrics with application to on-line signature recognition," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 40, no. 3, pp. 525–538, 2010.
- [96] C. Gruber, T. Gruber, S. Krinninger, and B. Sick, "Online signature verification with support vector machines based on lcss kernel functions," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 4, pp. 1088–1100, 2010.
- [97] R. Martens and L. Claesen, "On-line signature verification by dynamic time-warping," in *Proceedings of 13th International Conference on Pattern Recognition*, vol. 3, pp. 38–42 vol.3, 1996.
- [98] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using kinect sensor," *IEEE Transactions on Multimedia*, vol. 15, no. 5, pp. 1110–1120, 2013.
- [99] G. Marin, F. Dominio, and P. Zanuttigh, "Hand gesture recognition with leap motion and kinect devices," *2014 IEEE International Conference on Image Processing, ICIP 2014*, pp. 1565–1569, 01 2015.
- [100] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota, "Fingerio: Using active sonar for fine-grained finger tracking," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, (New York, NY, USA), pp. 1515–1525, Association for Computing Machinery, 2016.

- [101] Z. Lu, X. Chen, Q. Li, X. Zhang, and P. Zhou, "A hand gesture recognition framework and wearable gesture-based interaction prototype for mobile devices," *IEEE Transactions on Human-Machine Systems*, vol. 44, no. 2, pp. 293–299, 2014.
- [102] Y. Zhang and C. Harrison, "Tomo: Wearable, low-cost electrical impedance tomography for hand gesture recognition," in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, (New York, NY, USA), pp. 167–173, Association for Computing Machinery, 2015.
- [103] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, and J. Yang, "A framework for hand gesture recognition based on accelerometer and emg sensors," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 41, no. 6, pp. 1064–1076, 2011.
- [104] J. Gummeson, B. Priyantha, and J. Liu, "An energy harvesting wearable ring platform for gestureinput on surfaces," in *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '14, (New York, NY, USA), pp. 162–175, Association for Computing Machinery, 2014.
- [105] M. Roshandel, A. Munjal, P. Moghadam, S. Tajik, and H. Ketabdar, "Multi-sensor finger ring for authentication based on 3d signatures," in *Human-Computer Interaction. Advanced Interaction Modalities and Techniques* (M. Kurosu, ed.), (Cham), pp. 131–138, Springer International Publishing, 2014.
- [106] G. Wu, J. Wang, Y. Zhang, and S. Jiang, "A continuous identity authentication scheme based on physiological and behavioral characteristics," *Sensors (Basel, Switzerland)*, vol. 18, p. 179, Jan 2018. 29320463[pmid].
- [107] A. Levy, B. Nassi, Y. Elovici, and E. Shmueli, "Handwritten signature verification using wrist-worn devices," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, Sept. 2018.
- [108] G. Li and H. Sato, "Handwritten signature authentication using smartwatch motion sensors," in *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, pp. 1589–1596, 2020.
- [109] A. Laghari, Waheed-ur-Rehman, and Z. A. Memon, "Biometric authentication technique using smartphone sensor," in *2016 13th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pp. 381–384, 2016.
- [110] L. Ardüser, P. Bissig, P. Brandes, and R. Wattenhofer, "Recognizing text using motion data from a smartwatch," in *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pp. 1–6, 2016.

- [111] C. Xu, P. H. Pathak, and P. Mohapatra, "Finger-writing with smartwatch: A case for finger and hand gesture recognition using smartwatch," in *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, HotMobile '15, (New York, NY, USA), pp. 9–14, Association for Computing Machinery, 2015.
- [112] T. Zhao, J. Liu, Y. Wang, H. Liu, and Y. Chen, "Ppg-based finger-level gesture recognition leveraging wearables," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pp. 1457–1465, 04 2018.
- [113] A. Savitzky and M. J. E. Golay, "Smoothing and differentiation of data by simplified least squares procedures.," *Analytical Chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.
- [114] R. W. Schafer, "What is a savitzky-golay filter? [lecture notes]," *IEEE Signal Processing Magazine*, vol. 28, no. 4, pp. 111–117, 2011.
- [115] C. Truong, L. Oudre, and N. Vayatis, "Selective review of offline change point detection methods," *Signal Processing*, vol. 167, p. 107299, 2020.
- [116] Z.-L. Zhang and Z. Yi, "Robust extraction of specific signals with temporal structure," *Neurocomputing*, vol. 69, no. 7, pp. 888 – 893, 2006. New Issues in Neurocomputing: 13th European Symposium on Artificial Neural Networks.
- [117] M. Raghuram, K. Madhav, E. Krishna, N. R. Komalla, and K. Reddy, "A novel approach for motion artifact reduction in ppg signals based on as-lms adaptive filter," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, pp. 1445–1457, 2012.
- [118] M. Christ, N. Braun, J. Neuffer, and A. Kempa-Liehr, "Time series feature extraction on basis of scalable hypothesis tests (tsfresh - a python package)," *Neurocomputing*, 05 2018.
- [119] M. Christ, N. Braun, A. Kempa-Liehr, and J. Neuffer, "Overview on extracted features." Available at https://tsfresh.readthedocs.io/en/latest/text/list_of_features.html (30/11/2020).
- [120] Y. Benjamini and D. Yekutieli, "The control of the false discovery rate in multiple testing under dependency," *Ann. Stat.*, vol. 29, 08 2001.
- [121] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [122] J. L. Alexander, "The role of resistance exercise in weight loss," *Strength and Conditioning Journal*, vol. 24, pp. 65–69, 2002.

- [123] D. Rand, R. Kizony, U. Feintuch, N. Katz, N. Josman, and A. Rizzo, “Comparison of two vr platforms for rehabilitation: Video capture versus hmd,” *Teleoperators and Virtual Environments - Presence*, vol. 14, pp. 147–160, 04 2005.
- [124] C. Mattmann, O. Amft, H. Harms, G. Tröster, and F. Clemens, “Recognizing upper body postures using textile strain sensors,” *2007 11th IEEE International Symposium on Wearable Computers*, pp. 29–36, 2007.
- [125] A. B. M. M. Rahman, Y. Cao, X. Wei, P. Wang, F. Li, and Y. Wang, “PPGSign: Handwritten signature authentication using wearable PPG sensor,” in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2721–2726, 2022 © 2022 IEEE.